November 2016

Intent Recognition Of Rotation Versus Translation Movements In Human-Robot Collaborative Manipulation Tasks

Vinh Q. Nguyen Mr.
University of Massachusetts Amherst

Intent recognition of rotation versus translation movements in human-robot collaborative manipulation tasks

A Thesis Presented

by

Vinh Quang Nguyen

Submitted to the Graduate School of the

University of Massachusetts Amherst in partial fulfillment

of the requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

September 2016

Mechanical and Industrial Engineering

Intent recognition of rotation versus translation movements in human-robot collaborative manipulation tasks

A Thesis Presented

by

Vinh Quang Nguyen

Approved as to style and content by:

_____

Frank C. Sup IV, Chair

_____

Joydeep Biswas

_____

Kourosh Danai

_____

Sundar Krishnamurty, Department head

Mechanical and Industrial Engineering

ACKNOWLEDGMENTS

I would like to give a big thanks to my thesis advisor, Prof. Frank Sup, who guided me with his patience, motivation, and dedication. Also, I would like to thank my lab-mates and friends, especially Qiandong Nie, for all their help in this project.

ABSTRACT

INTENT RECOGNITION OF ROTATION VERSUS TRANSLATION MOVEMENTS IN HUMAN-ROBOT COLLABORATIVE MANIPULATION TASKS

SEPTEMBER 2016

VINH QUANG NGUYEN

B.S.E.E., HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY, HANOI, VIETNAM

M.S.M.E., UNIVERSITY OF MASSACHUSETTS, AMHERST, USA

Directed by: Professor Frank C. Sup IV

The goal of this thesis is to enable a robot to actively collaborate with a person to move an object in an efficient, smooth and robust manner. For a robot to actively assist a person it is key that the robot recognizes the actions or phases of a collaborative tasks. This requires the robot to have the ability to estimate a person's movement intent. A hurdle in collaboratively moving an object is determining whether the partner is trying to rotate or translate the object (the rotation versus translation problem). In this thesis, Hidden Markov Models (HMM) are used to recognize human intent of rotation or translation in real-time. Based on this recognition, an appropriate impedance control mode is selected to assist the person. The approach is tested on a seven degree-of-freedom industrial robot, KUKA LBR iiwa 14 R820, working with a human partner during manipulation tasks. Results show the HMMs can estimate human intent with accuracy of 87.5% by using only haptic data recorded from the robot. Integrated with impedance control, the robot is able to collaborate smoothly and efficiently with a person during the manipulation tasks. The HMMs are compared with a switching function based approach that uses interaction force magnitudes to recognize rotation versus translation. The results show that HMMs can predict correctly when fast rotation or slow translation is desired, whereas the switching function based on force magnitudes performs poorly.

CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Project overview

There has been increasing demand for robots to collaborate with humans, in which robots and humans share the same tasks and workspaces. A critical requirement is making the robots to interact safely and compliantly with humans in performing manipulation tasks. When position or force control alone cannot satisfy the above requirement due to interaction between the robot and the environment, a standard approach is impedance control (Al-jarrah & Zheng, 1997)(Lawitzky, Mortl, & Hirche, 2010). Impedance control enables a robot to react compliantly while stabilizing an object so that a person can relay direction intent and move it freely using less effort as depicted in the scenario in Figure 1.

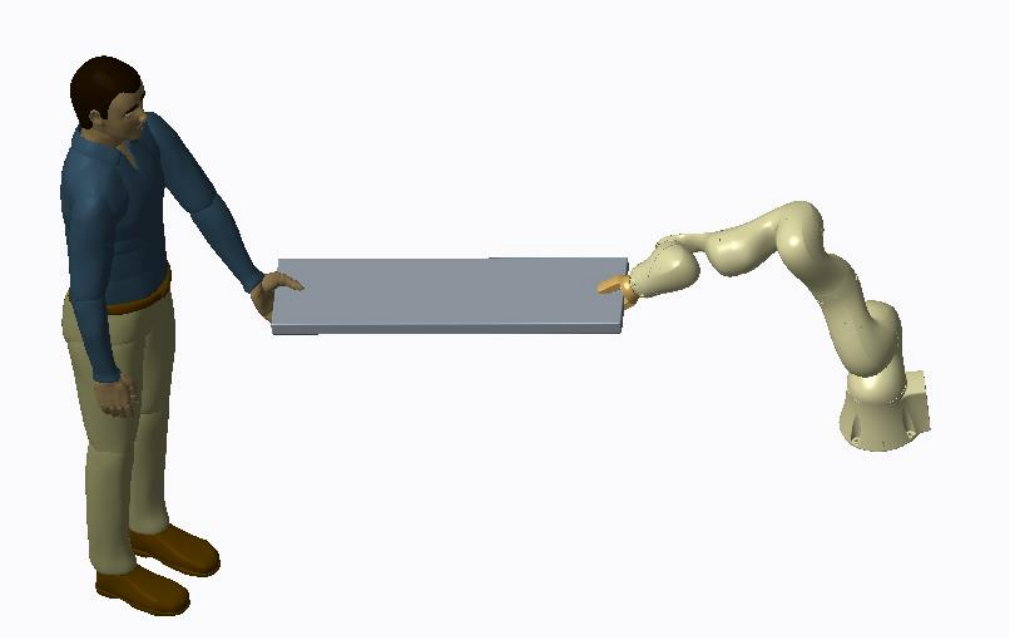


Figure 1: A human robot cooperative manipulation task

However, impedance control alone cannot fulfil the task of manipulating object when the person is unable to apply large torque to rotate the object (Arai, Takubo, Hayashibara, & Tanie, 2000). This can happen when the person holds the object with one hand or the object is significant long. The force applied by the person is mainly perpendicular to the object (Arai et al., 2000). In this scenario, both rotation and translation intent cause the object to rotate and translate at the same time, which makes the manipulation tasks difficult to achieve. The key is how a robot is able to distinguish whether the person is trying to rotate the object or translate it laterally, so that it can react appropriately.

There are two ways to address this problem. One is to apply additional constraints to the mobility of the robot or the interpretation of sensor data to get an unequivocal interpretation of the measured data; the other is to estimate the human partner's movement intention and then react appropriately (Yigit, Burghart, & Worn, 2004). An example of the former approach is virtual non-holonomic constraints (Arai et al., 2000). When moving an object like a wheelbarrow any forces perpendicular to the object are interpreted as rotation and only translation along the object's length direction is allowed. Although this approach showed that the object can be moved to any position, it requires the operator to combine a series of actions in order to perform a movement and is not suitable when working in a restricted working space (Wojtara et al., 2009). Another example is to use additional sensors to get an unequivocal interpretation. In (Wojtara et al., 2009), the rotation is interpreted by the torque around the person's gripping point; therefore an additional force sensor between the person's handle and the object is needed. Typically, this is not possible or desired to require additional sensing between the user and the object.

The second method tries to switch between rotation and translation mode based on the estimation of the person's intention. In early work, Yokoyama et al. [2003] proposed a voice recognition for switching which was operated by allowing a person to verbally direct a robot to change from rotation to translation mode. For this approach, an advanced speech and language recognition was required and

the communication interface significantly reduces the interaction speed. In addition to verbal cues, haptic feedback plays the most important role in collaborative manipulation (Feth, Groten, Peer, Hirche, & Buss, 2009). Therefore, some research used haptic data to make the transition between two modes (rotation and translation) automatic. Dumora et al. [2013] tried to solve a the problem of intent recognition base on haptic and position data using a naïve Bayes classifier which is a supervised learning algorithm. However, an additional force and torque sensor was used between the handle where the human grasped the object and the object itself. It measured the torque applied on the object. In the real world, this is not possible configuration.

Another example for solving the rotation and translation problem by mode switching is in (Karayiannidis, Smith, & Kragic, 2014). The authors use force direction and magnitude to switch between two modes using switching functions using preselected thresholds. While this approach is straight forward to implement, it has some limitations. First, using force direction condition has problems with chattering as the number of mode switching is several hundred times during a simple semicircular motion. Second, better performance is found when adjusting the force magnitude; however, this has a problem in the case of fast rotations and slow translations. When fast rotation is desired, the force magnitude is above the switching threshold and the controller will infer as translation mode. Similarly, in slow translation, force magnitude is below the switching threshold so the controller is likely to infer the rotation mode.

In summary, the previous works tried to address the problem by either adding constraints into the mobility of the robot or using additional torque sensor at the person's gripping point. There has been only one research tried to solve the problem without adding constraints and only use haptic data from robot's sensors (Karayiannidis et al., 2014). However, it has the problem when fast rotation or slow translation is desired. Also, a point to notice in previously mentioned works is that the algorithms are based on the static conditions. Only the instantaneous data at the current state are used and considered for recognition of the users' intent. However, it would be more natural and potentially successful to use

dynamics analyses with time series data (Takeda, Kosuge, & Hirata, 2005). There are several methods which can be used with time series data such as Hidden Markov Models, Dynamic Time Warping, Recurrent Neural Network. In (Takeda et al., 2005), results showed Hidden Markov Models outperformed Neural Network in the case of intent recognition when working with haptic data. Also, in those methods, Hidden Markov Models have been widely and successfully used for speech recognition, gesture recognition, and intent recognition. For this reason, this project evaluates HMMs working with time series data for intent recognition of the rotation versus translation problem.

1.2 Research Objective

In this thesis, Hidden Markov Models (HMMs) are used to solve the problem of recognizing collaborative rotation versus translation movements during the person holding the object with one hand. HMMs have been used for human intent recognition based on the analysis of speech, visual, and haptic data (Huang, 1992) (Starner, 1995) (Chen, Fu, & Huang, 2003)(N Stefanov, Peer, & Buss, 2010) (Wang, Peer, & Buss, 2009)(Kucukyilmaz, Sezgin, & Basdogan, 2013). However, there is no research can be found using this approach to solve the rotation versus translation problem in a human-robot collaborative manipulation task. In this project, the robot will distinguish a person's intent of trying to rotate or translate an object collaboratively using HMMs. Different from most of previous works, in this project, the person holds the object with only one hand and only haptic data from the robot's side is used. That means there is not an additional force and torque sensor at the person's grasp point. After recognition, the robot will then actively contribute to the task. Impedance control is chosen for safe and compliant interaction purpose. An experiment with an industrial robot (KUKA LBR iiwa 14 R820) working with a person in moving a weighted wooden bar is conducted. The results shows the proposed method enables communication of intent from the human to the robot. For comparison, the proposed method is baselined against switching function method based on force magnitude (Karayiannidis et al., 2014).

1.3 Thesis organizations

This thesis is divided into five chapters. This first chapter introduces the motivations and scope of the project. The second chapter reviews the relevant background on human-robot collaboration in manipulation tasks, impedance control, and intent recognition using HMM. Also discussed in this chapter are related examples of applications of impedance control and HMM in human-robot collaboration. The third and fourth chapter presents the proposed method and the experimental results. The fifth chapter presents a discussion of the results. Finally, the last chapter summaries conclusions of this project and opens more ideas for future works and development.

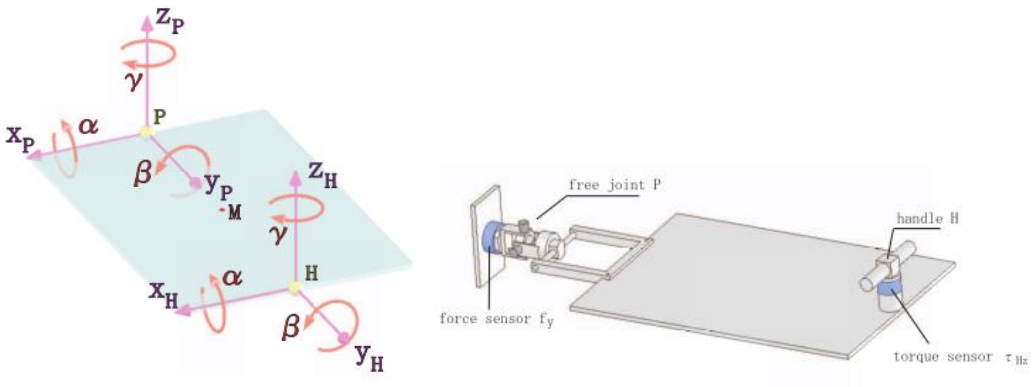
CHAPTER 2

BACKGROUND

2.1 Prior works: The rotation versus translation problem

An early work on the rotation versus translation problem by Yokoyama et al. used a human interface based on voice instruction to help the users instruct the robot to move “Right”, “Left”, “Clockwise”, “Anticlockwise” (Yokoyama et al., 2003). For this approach, an advanced speech and language recognition is required. In addition, the communication interface significantly reduces the interaction speed. In addition to verbal cues, haptic feedback plays the most important role in collaborative manipulation (Feth et al., 2009). Some works implemented different approaches with haptic data to have the robot automatically switch between rotation and translation modes. This section reviews relevant works which used haptic data to solve the rotation versus translation problem in collaborative manipulation tasks.

In (Wojtara et al., 2009), the authors implemented the partner-that-follows algorithm, in which the torque information is used to control the rotation mode, and the displacement information is used to control the translation mode. In this case, the robot is the follower and the human is the leader.



(a) (b)

Figure 2: Coordinate system of object and object holding frame (Wojtara et al., 2009)

Two coordinate systems were put at gripping point of robot (P) and gripping point of the human (H) (Figure 2). In this work, instead of separating the rotation and translation modes, the robots move passively with the haptic interaction from the person. The control law for velocity in x direction v_{Px} and velocity in z direction v_{Pz} was based on position information:

$$\begin{aligned} v_{Px} &= c_x\gamma - c_\tau \frac{d}{dt}\cos\beta \\ v_{Pz} &= c_z\alpha + c_\tau \frac{d}{dt}\sin\beta \end{aligned} \tag{1}$$

c_x, c_z, c_τ are constant control parameters that were chosen manually according to satisfy the user's feeling; γ, β are angles around z_H, y_H axes. The control law for the force in y direction is

$$f_{Py} = \frac{dy}{dt} d_{yR} + m_{yR}\tau_{Hz} \tag{2}$$

With d_{yR} is the damping constant, m_{yR} is the mass constant of impedance control. Experiments were carried out with three subjects moving the object from one position and placing it exactly in the target position. The result of implementing the partner-that-follows algorithm showed high precise in placing the object.

Dumora et al. tried to solve a similar problem of intent recognition base on haptic and position data (Dumora, Geffard, Bidard, Aspragathos, & Fraisse, 2013). A naïve Bayes classifier which is a supervised learning algorithm based on Bayes' theorem was chosen. If there are m overall possible intentions $C = \{C_1, C_2, ..., C_m\}$, and n haptic measured $F = \{F_1 = f^1, ..., F_n = f^n\}$, then the prediction of intentions is based on the probability:

$$\Pr(C = C_k \mid \bigcap_{i=1}^{n} F_i = f^i) = \frac{\Pr(C = C_k)\Pr(\bigcap_{i=1}^{n} F_i = f^i \mid C = C_k)}{\Pr(\bigcap_{i=1}^{n} F_i = f^i)} \quad (3)$$

The method consists of two steps. In the training step: using the dataset, the posterior probability $\Pr(C = C_k \mid \bigcap_{i=1}^{n} F_i = f^i)$ of each data point belong to each class was estimated by estimating the parameters of the likelihood $\Pr(\bigcap_{i=1}^{n} F_i = f^i \mid C = C_k)$ and the prior probability $\Pr(C = C_k)$. The next step is the prediction step, in which the class C^* is chosen as the result if it has the largest posterior probability higher than a confidence level γ .

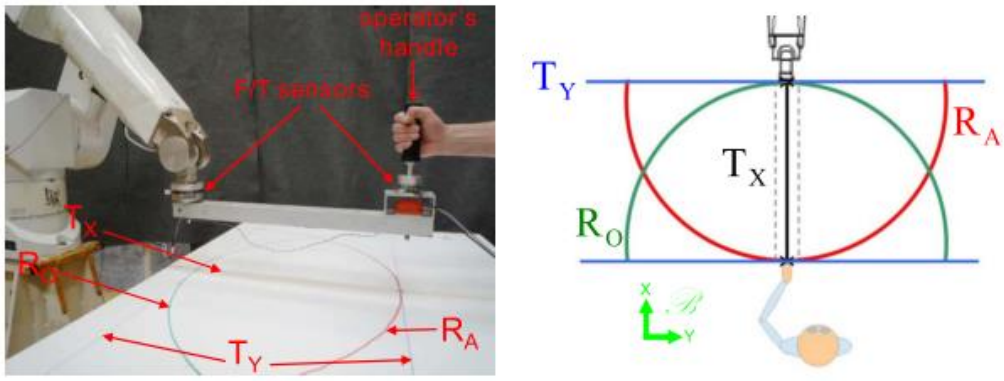


Figure 3: Experimental apparatus and drawn motions (Dumora et al., 2013)

The method was evaluated with experiments human robot collaboratively moving a bar (Figure 3). The experiments used a 6 axis force/torque sensor at wrist of robot arm and a second force/torque sensor at user's handle. The haptic features were the force along X-axis F_x , the torque along Z-axis M_z and the displacement of the user's handle in Y-axis ΔX_y. These values are expressed at the person's

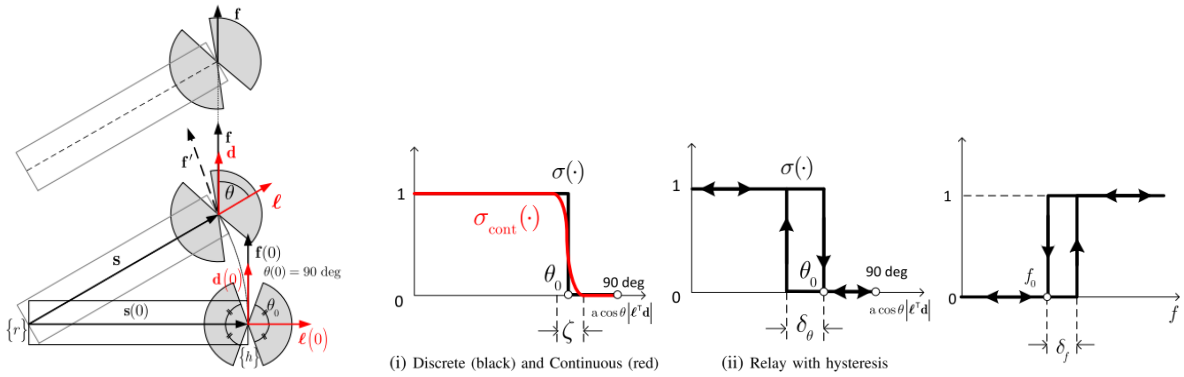
grasping point. Four different classes of motion were tested: T_Y - pure lateral translation, T_X - forward and backward translation, R_O rotation of the object around the user's gripping point, R_A - rotation of the object around the robot's gripping point (Figure 3). The results showed 92.8 % success rate using a confidence level $\gamma = 0.75$, and 97% when $\gamma = 0.9$.

One noticeable point in the two above works (Wojtara et al., 2009) and (Dumora et al., 2013) is that, a second force-torque sensor was placed between user's handle and the object, which was able to obtain a good measurement of the torque along Z-axis, τ_{Hz} at the person's gripping point. This is not the case in real world scenarios where there is no sensor between the handle and object. The scenario without this sensor is shown in (Figure 4) and investigated in (Karayiannidis et al., 2014).

Figure 4: Joint manipulation of an object by a robot and a person (Karayiannidis et al., 2014)

In their research, a person grasps the object with one hand which significantly limits the amount of torque a person can apply directly to the object. In addition, there was no direct measurement of force-torque at the contact point between the person and the object. In such a scenario, the previous methods rely on the torque measurements on the object to infer rotation or translation modes and are no longer

suitable. To solve the problem of rotation versus translation, the authors used the information of direction and magnitude of applied force which is measured at the robot's end-effector to switch between two modes. The first condition is based on the angle θ between the human force direction d and the virtual stick direction $\boldsymbol{\ell}$. When the angle $\theta > \theta_0$ the object will rotate, otherwise the system will switch to the translation mode (Figure 5a-b).



(a) (b) (c) (d)

Figure 5: Switching based on the force direction (a, b, c), switching function base on magnitude of torque (d) (Karayiannidis et al., 2014)

To reduce chattering, a smooth transition function was used to replace the sharp transition (red line in Figure 5b). Furthermore, to avoid switching due to force measurement noise, a relay term was added to the switching function (Figure 5c). The second condition is based on magnitude of applied force with a delay switching function (Figure 5d). Experiments were carried out with human and robot jointly held a 30 cm long wooden object to follow different paths (circular rotational motion, circular translational motion, straight-line translational motion). The results showed the robot was able to trigger switching conditions to select either rotation or translation mode. Better performance is when using force

magnitude condition as the number of mode switching are much less than using force direction. However, the performance using force magnitude may be questionable in the case of fast rotations and slow translations. When fast rotation is desired, the force magnitude is above the switching threshold and the controller will infer as translation mode. Similarly, in slow translation, force magnitude is below the switching threshold so the controller is likely to infer the rotation mode.

In summary, the previous works tried to address the problem by either constraining the mobility of the robot or using an additional torque sensor at the person's gripping point. There has been only one attempt at solving the problem without adding constraints and only using data from the sensors on the robot (Karayiannidis et al., 2014). However, it has problems detecting fast rotations or slow translations. Also, a significant issue with the previous works is that they only used instantaneous data from the current state to recognize the user's intent. However, it would be more natural and possibly more successful to use dynamics analyses with time series data (Takeda et al., 2005). There are several methods which can be used with time series data such as: Hidden Markov Models, Dynamic Time Warping, and Recurrent Neural Network. In (Takeda et al., 2005), results showed HMMs outperformed Neural Networks in the case of intent prediction when using force and torque data. Also, in those methods, HMMs have been successfully used for speech recognition, gesture recognition, and intent recognition. For these reasons, this project evaluates HMMs working with time series data for intent recognition of the rotation versus translation problem.

2.2 Hidden Markov Model

Hidden Markov Model (HMM) is a double stochastic process, one underlying Markovian stochastic process being not directly observable, but can only be observed through another stochastic process with a certain probabilistic of observation (Juang, 1986). HMM has been used successfully in speech

recognition (Huang et al., 1990) (Huang, 1992), visual recognition (Starner, 1995) (Chen et al., 2003), intent recognition based on haptic data (Takeda et al., 2005) (Wang et al., 2009) (N Stefanov et al., 2010).

Hidden Markov Model was derived from Markov chains theory which has been known since the 1980's (Juang, 1986). In real world process, as having a sequence of observation symbols which can be discrete (outcomes of coin tossing experiment, human name, etc.) or continuous (speech samples, temperature of a day, etc.), the problem is how to build a signal model that is able to explain and characterize the observed symbols. Hidden Markov Model tries to treat this problem under a probabilistic or statistical framework.

Let's take an example of a stochastic process (Figure 6): Urns and Balls (Rabiner, 1989). In this example, there are n urns. Each urn has a large number of color balls, and there are m possible colors. Now, a person will randomly choose an urn and then randomly pick a ball from that urn. They record the color of the ball and then returns the ball back to the urn. They keep doing this to form a sequence of colors of balls they picked $O = \{\text{Green}, \text{Green}, \text{Blue}, \text{Red}, ..., \text{Blue}\}$. In this example, each urn could be modeled as one state, so there are n states and described by a set of states $S = \{s_1, s_2, ..., s_n\}$. There are m possible output symbols which are colors of balls. The probabilities that the person moved from urn i^{th} to the urn j^{th} are described by transition probability a_{ij}. Here, the probability of changing from a state s_i at time t to another state s_j only depends of the current state s_i. This is the Markov property. The probabilities a color is picked when the person stopped at one urn are described by the output symbol probabilities $b_j(k)$ (with $k = 1, 2, ..., m$).

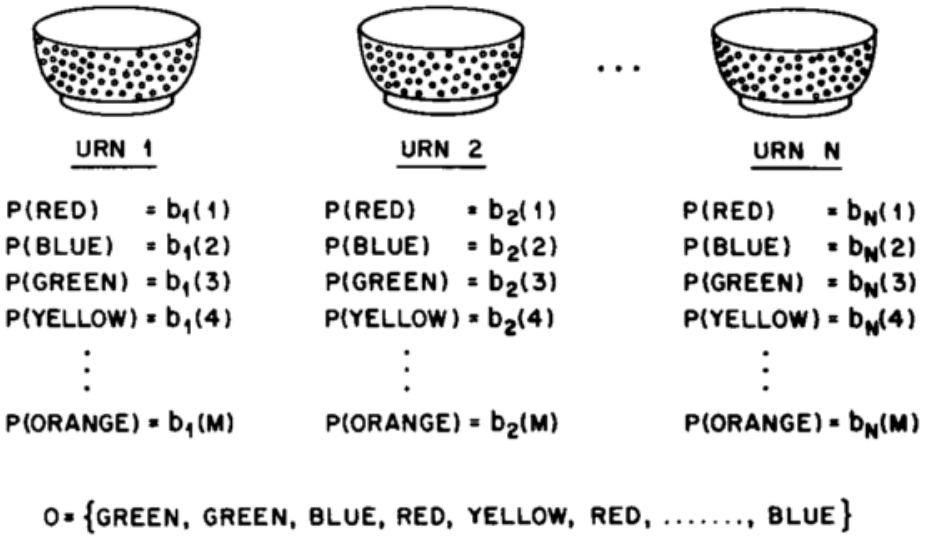


Figure 6: Urns and Ball example (Rabiner, 1989)

Now, the process can be model with a Hidden Markov Model. Given a set of states $S = \{s_1, s_2, ..., s_n\}$, the initial state distribution $\pi = \{\pi_i\}$ the set of output symbols $V = \{v_1, v_2, ..., v_m\}$, there are two set of probabilities need to consider: a transition probability and output probability distribution (Juang, 1986). The transition probability $A = \{a_{ij}\}$ defines the probability of a new state S_j is entered given a current state s_i . The output probability distribution $B = \{b_j(k)\}$ for the case discrete output defines the condition probability of emitting output symbol v_k given the current state S_j. An example of a Hidden Markov Model with three states is shown in Figure 7.

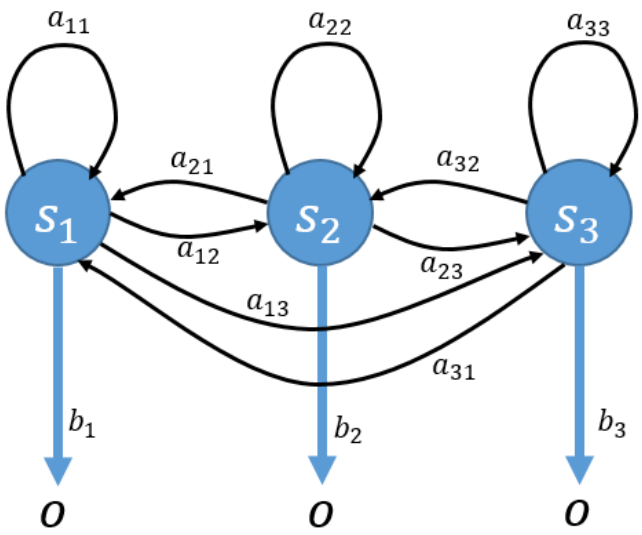


Figure 7: Example of Hidden Markov Model

A complete parameter set of a discrete output symbol HMM can be expressed as:

$$\lambda = (A, B, \pi) \tag{4}$$

Symbols	Meaning
$S = \{s_1, s_2, ..., s_n\}$	Set of n states.
$V = \{v_1, v_2, ..., v_m\}$	Set of m output symbols.
$O = \{o_k\}$	discrete set of possible symbol observations
$A = \{a_{ij}\}$, $a_{ij} = \Pr(S_j \mid S_i)$	State transition probability distribution
$B = \{b_j(k)\}$, $b_j(k) = \Pr(v_k \mid S_j)$	Observation symbol probability distribution
$\pi = \{\pi_i\}$, $\pi_i = \Pr(S_i$ is the first state $)$	Initial state distribution

Table 1: Meaning of symbols in HMMs

In the case of continuous output $\mathbf{x}$, the observation symbol probability distribution is replaced by $B = \{b_j(\mathbf{x})\}$, where $b_j(\mathbf{x})$ is now the probability density.

$$b_j(\mathbf{x}) = \sum_{m=1}^{M} c_{jm} \Im(\mathbf{x}, \mu_{jm}, U_{jm})$$

where $\mathbf{x}$ is the vector modeled; c_{jm} is the mixture coefficient for m^{th} mixture in state j; $\Im$ is usually chosen as Gaussian density function; μ_{jm} is the mean vector and U_{jm} is the covariance of m^{th} mixture component in state j.

Given a HMM form, there are three basic problems of interest that need to be solved for the model to be useful in the real world applications (Juang, 1986):

1. Given the model $\lambda = (A, B, \pi)$ and the observation sequence $O = \{o_1, o_2, ..., o_T\}$, compute the probability of observation sequence $\Pr(O, \lambda)$.
2. Given the model $\lambda = (A, B, \pi)$ and the observation sequence $O = \{o_1, o_2, ..., o_T\}$, how to find an optimal state sequence $S = \{s_1, s_2, ..., s_T\}$ which can generate the observation sequence in some meaningful sense.
3. Given some observation sequences, how to adjust the model parameters of $\lambda = (A, B, \pi)$

The first problem is the evaluation problem, it is useful to determine which model best matches the observations. To solving this problem, forward-backward procedure is implemented. The second problem tries to uncover the hidden state sequence or states (e.g. the intention state of human, the object in picture, etc.), namely finding the "optimal" state sequences associated with the given observation sequence. A technique for finding this best state sequence based on dynamic programming methods is called Virterbi algorithm. In the third problem, the model parameters are optimized so that it can best describe how the observation sequence is generated. It is the training process for the model. For this problem, an iterative procedure, Baum-Welch method is implemented.

To use HMM for human intent recognition, research has used haptic data for the model. (N Stefanov et al., 2010) analyzed the interaction force signals from human hand to detect intent of transporting or positioning an object. The authors proposed a method of extracting online and event

based features from data. For training the model, 30 training datasets are used for each phase of intent to be detected. The accuracy of overall predicting is 70.73 %, 96.35% for the positioning task and 59.6% for transportation task. The lower accuracy in transportation tasks was explained by similar human behavior and thus observed patterns in positioning and transportation tasks when assistance was provided. The results also showed that the HMM enabled a very fast recognition when using haptic data (Figure 8).

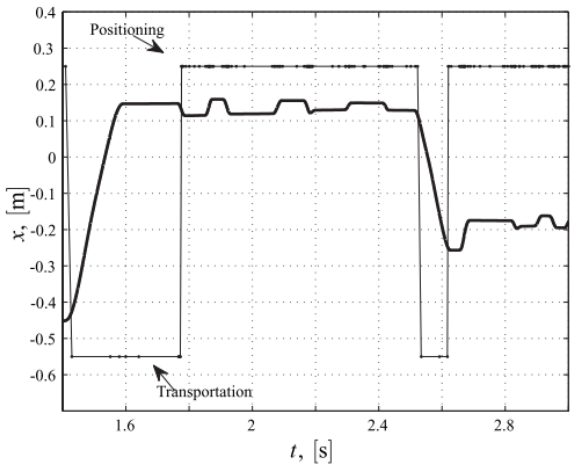


Figure 8: Results for intent recognition using HMMs (N Stefanov et al., 2010)

Takeda et al. used HMMs for the robot which danced with a person to predict the next dance step intended by the person (Takeda et al., 2005). For each dance step, a time series data with a length $T_{effective}$ of force/torque applied by the human dancer is used. This data series are latter discretized with a time segment ΔT . With the proposed system, the robot successfully estimated the next dance step of human intent. The result showed higher accuracy using HMMs comparing to using a neural network model. These above works show that using just measured haptic data, HMMs are able to estimate very well and fast the person's intentions. Yet, this approach has not been applied to the rotation versus translation problem in collaborative manipulation tasks. Therefore, this thesis focuses on solving the intent recognition problem during human robot collaborative tasks using HMMs. After the intents are recognized, the robot

need to react appropriately in a compliant and safe way. For this requirement, impedance control is implemented.

2.3 Impedance control

In human-robot interaction, it is expected that a manipulator should be able to adapt to the physical changes in the environment while still keeping the desired motion. In other words, the controller needs to control its motion (such as following trajectory – position control) and in addition modulate it in response to a disturbance from the environment. Position or force control alone cannot satisfy the above requirements. The reason is the interaction between manipulator and environment affects the controlled variables. The resulting error impacts performance or may cause the system to become unstable (Buerger & Hogan, 2007). Another approach known as disturbance rejection considers the environment's dynamics as disturbances. The performance of this approach depends on bounding the disturbances force. However, in this case the interaction force may be very large or even exceed the robot's nominal capacity. Therefore, this approach does not seem promising.

Another approach is to model the environment as an uncertain part of the robot and use robust control. In the robust control approach, the system uses a controller that is able to work with changes of parameters of the system within a range. During interaction with the environment the range of parameters may be too large and this type of controller might sacrifice performance too much. Thus, treating interaction as a robustness problem is not suitable (N Hogan & Buerger, 2005). Adaptive control is able to expand the range of uncertainty for which performance specification can be achieved. In adaptive control, one needs to do online estimation of the uncertain parameters then applies a controller that guarantees some desired behavior of the system. A limitation of this approach is that the estimation procedure requires excitation of the system dynamics which may not be desired. Also, for systems with possibly fast varying parameters, adaptive control is not applicable (Yi & Zhang, 2001). Therefore, adaptive

control is not promising for the case of interaction control. Another reason that makes independent control of force and motion not good is that because the environment can impose kinematic constraint or dynamic constraint. For example, when the manipulator contacts a rigid surface, this is a kinematic constraint. In this case, a motion controller cannot work. Similarly, force control alone cannot work if there is not contact between the manipulator and the environment.

Impedance control was developed to overcome these issues (Neville Hogan, 1985). The idea is to treat the environment as admittance and the manipulator as an impedance. It is derived from the postulate, "it is impossible to devise a controller which will cause a physical system to present an apparent behavior to its environment which is distinguishable from that of a purely physical system". Thus, a controlled system is considered an equivalent physical system. Now, the manipulator and the environment create two physical systems, in which the instantaneous power flow between them is defined as the product of two conjugate variables: an effort and flow. Examples of effort could be force or voltage, while flow could be velocity or current. The environment usually contains inertias or kinematic constraints which accepts effort as an input and yields flow as an output – thus is considered admittance; meanwhile, the manipulator accepts flow as input and yields effort as the output – thus is considered impedance. Different from force or position control, impedance control does not try to regulate force or position, instead it regulates the relationship between effort and flow (force and velocity in this thesis) (Figure 9). The nodic impedance defines the desired interaction dynamics or the relationship between effort and flow. This relationship is usually described by a function of three parameters: mass m , damping b and stiffness k .

$$F = k(V_0 - V) + b(\dot{V}_0 - \dot{V}) + m(\dot{V}_0 - \dot{V}) \tag{5}$$

The virtual trajectory V_0 is similar to a reference trajectory of motion control, but there is no assumption that the dynamic of virtual trajectory must be slower than the servo dynamic. In addition, the virtual

trajectory need not to be realizable. For example, it need not to be confined to manipulator's workspace (Hogan, 2011).

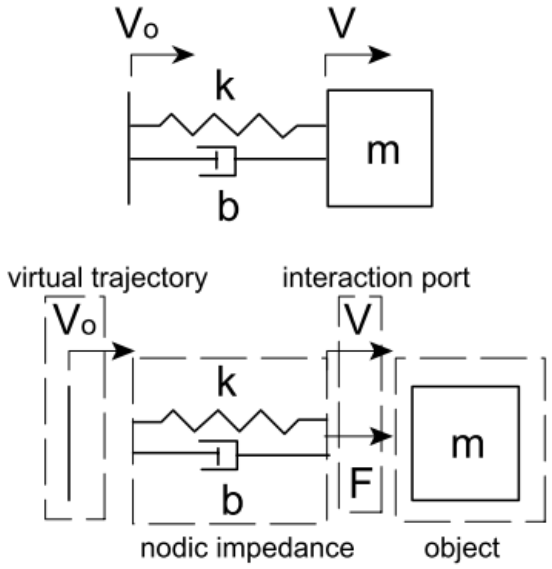


Figure 9: Desired interaction relationship (Hogan, 2011)

Hogan proposed a simple approach to implement impedance control. Consider a multi degree-of-freedom manipulator, a simple impedance controller is follow:

$$\mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{D}(\dot{\boldsymbol{\theta}}) = \mathbf{T}_{act} + \mathbf{T}_{\text{int}} \tag{6}$$

Where $\mathbf{M}$ is the inertia matrix, $\boldsymbol{\theta}$ is the vector of joint angles, $\mathbf{C}$ is the inertia coupling torques matrix (due to Coriolis/ centrifugal accelerations), $\mathbf{D}$ is the vector of dissipative torques due to friction, $\mathbf{T}_{act}$ is the vector of actuated torques, $\mathbf{T}_{\text{int}}$ is the vector of environment torques. If the manipulator behaves like a spring with a stiffness $\mathbf{K}_j$ and a damper $\mathbf{B}_j$, then the control law is:

$$\mathbf{T}_{\mathbf{act}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \mathbf{K}_{\mathbf{j}}(\boldsymbol{\theta}_{\mathbf{0}} - \boldsymbol{\theta}) + \mathbf{B}_{\mathbf{j}}(\dot{\boldsymbol{\theta}}_{\mathbf{0}} - \dot{\boldsymbol{\theta}}) \tag{7}$$

With $\boldsymbol{\theta}_0$ and $\dot{\boldsymbol{\theta}}_0$ are desired trajectory and velocity. Combining (6) and (7), the equation for impedance control:

$$\mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) + \mathbf{D}(\dot{\boldsymbol{\theta}}) + \mathbf{B}_\mathbf{j}\dot{\boldsymbol{\theta}} + \mathbf{K}_\mathbf{j}\boldsymbol{\theta} = \mathbf{K}_\mathbf{j}\boldsymbol{\theta}_\mathbf{0} + \mathbf{B}_\mathbf{j}\dot{\boldsymbol{\theta}} + \mathbf{T}_{\text{int}} \tag{8}$$

The stiffness and damping matrices $\mathbf{K}_j$ and $\mathbf{B}_j$ result in stiffness and damping at the end-effector of the manipulator, but they vary with the changes in the position of the robot. Therefore, to have stiffness k and damping b at the end effector, a transformation from joint space to end-effector space is needed. The position and velocity of end-effector $\mathbf{x} = \mathbf{L}(\boldsymbol{\theta})$, $\dot{\mathbf{x}} = \mathbf{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}$. The control law is as follows:

$$\mathbf{T}_{\mathbf{act}}(\mathbf{x}_0, \dot{\mathbf{x}}_0, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \mathbf{J}^T(k(\mathbf{x}_0 - \mathbf{L}(\boldsymbol{\theta})) + b(\dot{\mathbf{x}}_0 - \mathbf{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}) \tag{9}$$

In this approach, the controller does not use force or torque feedbacks and, thus, does not require a force-torque sensor. The controller will well behave if the manipulator's inertia and friction are relatively small. With considerable friction and/or inertia, the above simple impedance control technique may not provide good performance as the controller does not compensate for these physical impedances. A solution for this is to use force feedback that minimizes the deviation of the actual end-effector force from the desired value (Buerger & Hogan, 2007). So far, impedance control is widely used in the field of robotics, especially in human-robot interaction, control of prosthesis and exoskeleton, or robots which interact with the environment. In the field of human-robot interaction, impedance control is usually used to make the robots have compliant interaction with human. This is important to enable safe, natural and stable between both the human and robot when sharing the same workspace.

Impedance control has been used on robots to help humans during an object manipulation task. Lawitzky et al. combined different effort sharing policies with a hierarchical impedance based motion control framework for human-robot collaboration in moving object task (Lawitzky et al., 2010). At lower

levels, the robot generates force based on the effort sharing policy. At higher levels, a closed loop impedance control scheme modifies the force references for tolerance towards unexpected human behavior. Experiment results proved human force inputs decrease by more than 50% in the direction of effort sharing when maximizing the robot's effort. The tracking performance also improved in the direction of effort sharing when increasing the robot's effort.

In (Gribovskaya, Kheddar, & Billard, 2011), authors proposed an adaptive impedance controller that was able to compensate for non-modeled effect. The system's performance was evaluated in controlled situations using physical simulation of a pair of planar robot in which one robot played as the leader and the other was the follower. The simulation results showed the system was stable and able to continuously adapt to the incoming force.

In (Yokoyama et al., 2003), authors applied an impedance controller based on force sensing for a humanoid robot HRP-2P. The experiments were carried out with the robot working cooperatively with a person to move a panel from loading zone to some other places. By using impedance control, the transmitted force on the robot by the human is absorbed smoothly, hence avoiding jerks created to tip over the robot.

In (Rozo et al., 2013) a KUKA light-weight 7 degree-of-freedom robot and a person collaboratively in a table assembling task. The robot's role is to hold a table when a person screws the four legs to it. The robot is controlled by impedance control with the impedance parameters are learned via a second person's demonstration. After learning, the task can be reproduced by a single user with the help of the robot.

In above examples, impedance control plays important role to make the robot react compliantly and robustly with humans and environments. However, impedance control alone cannot fulfil the manipulating task when human and robot collaborate in moving a long object (Arai et al., 2000).

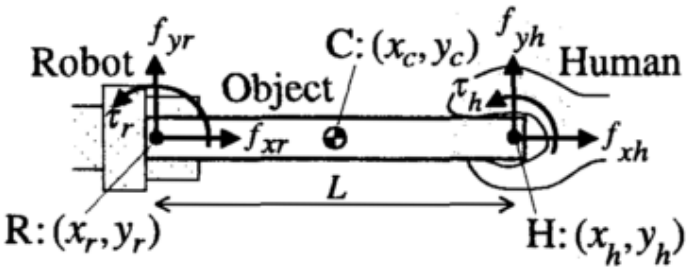


Figure 10: Collaborative manipulation with long object (Arai et al., 2000)

Consider a case when a person and a robot jointly hold a long wooden bar. The person holds the object of mass M at point H, and the robot holds at point R (Figure 10). As the object moves in a horizontal plane, the object's movement is described by:

$$\begin{cases} M\ddot{x}_c = f_{xh} + f_{xr} \\ M\ddot{y}_c = f_{yh} + f_{yr} \\ I\ddot{\theta} = \dfrac{f_{yh}L}{2} - \dfrac{f_{yr}L}{2} + \tau_h + \tau_r \end{cases} \tag{10}$$

An impedance behavior comprising of mass m and viscous friction b will be described by

$$\begin{cases} -f_{xr} = m\ddot{x}_r + b\dot{x}_r \\ -f_{yr} = m\ddot{y}_r + b\dot{y}_r \\ -\tau_r = i\ddot{\theta}_r + c\dot{\theta}_r \end{cases} \tag{11}$$

Here $\theta = \theta_r$ is very small when the sampling frequency is high enough, so the kinematic relationship between the center of mass C and the robot gripping point R:

$$\begin{cases} \ddot{x}_c = \ddot{x}_r - \dfrac{L}{2}\dot{\theta}^2 \\ \ddot{y}_c = \ddot{y}_r + \dfrac{L}{2}\ddot{\theta} \end{cases} \tag{12}$$

Replace Equations (11) and (12) into (10) results in:

$$\begin{cases} f_{xh} = (M+m)\ddot{x}_r + b\dot{x}_r - \dfrac{ML\dot{\theta}^2}{2} \\ f_{yh} = (M+m)\ddot{y}_r + b\dot{y}_r + \dfrac{ML\ddot{\theta}}{2} \\ \tau_h = (I+i-\dfrac{ML^2}{4})\ddot{\theta} - (\dfrac{M}{2}+m)L\ddot{y}_r + c\dot{\theta} - bL\dot{y}_r \end{cases} \tag{13}$$

The motion in x - axis corresponds directly with the f_{xh} . Meanwhile, the normal acceleration $\ddot{y}_r$ and angular acceleration $\ddot{\theta}$ are coupled. With the case of long object, it is hard to apply a large torque at the end of the object, so here τ_h is assumed to be equal to zero. Now, motion of the object in rotation and translation is given by equations (Arai et al., 2000):

$$\begin{cases} \ddot{y}_r = \dfrac{(I+i-\dfrac{ML^2}{4})f_{yh} - (I+i+\dfrac{ML^2}{4})b\dot{y}_r + \dfrac{MLc\dot{\theta}}{2}}{(M+m)(I+i)+\dfrac{MmL^2}{4}} \\ \ddot{\theta} = \dfrac{(\dfrac{M}{2}+m)Lf_{yh} + \dfrac{MLb\dot{y}_r}{2} - (M+m)c\dot{\theta}}{(M+m)(I+i)+\dfrac{MmL^2}{4}} \end{cases} \tag{14}$$

Both the person's intention of either rotation or translation the object cause the torque around z axis at the robot's gripping point R , this leads to sideslip of the object and complicates the manipulation (Arai et al., 2000). There are two principal ways to address this problem: first is applying additional constraints to the mobility of the robot or the interpretation of sensor data to get an unequivocal interpretation of the measure data, second is to estimating the human partner's intent of movement then react according to (Yigit et al., 2004). An example of the former was a virtual non-holonomic constraints proposed in (Arai et al., 2000). In the movement of an object like a wheelbarrow any forces perpendicular to the object are interpreted as rotation and only translation along the x axis is allowed. Although this approach has shown

to be able to move the object to any position, it would require the operator to combine a series of motion in order to perform a single movement. Also, it is not suitable in cases of a restricted workspace (Wojtara et al., 2009). Therefore, in this thesis the approach of a human intent recognition during collaborative manipulation is the focus.

CHAPTER 3

METHOD

This thesis focuses on human-robot collaboration of manipulation object task where a person and a robot jointly move an object. The robot is holding the object, a weighted board, and the person grasps the other end of the object with one hand (Figure 11). The desired path trajectory is not provided to the robot and the robot is not aware in advance whether the person intends to rotate or translate the object. The robot will rely on forces and torques measured by the robot at the gripping point of the robot to predict the person's intended motion in real-time. HMMs are trained with a measured dataset and then are used for the robot to estimate the person's intent of rotation or translation the object (details in section 3.1). After that, the robot will move to follow the person's intention. An impedance controller is implemented to the robot for safe and compliant interactions.

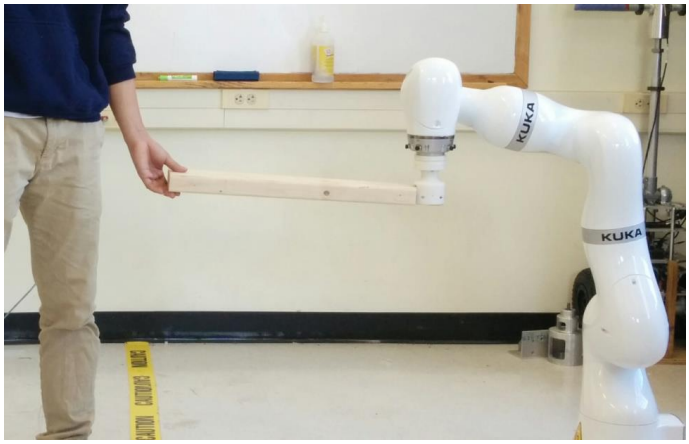


Figure 11: Human-robot collaboration in manipulation an object

3.1 Robot testbed

For the experiments, a KUKA LBR iiwa 14 R820 robotic arm is used (Figure 11). This is a light weight, 7 degree-of-freedom robot arm with a rated load of 14 kg and a total weight of 29.6 kg. It is the latest

version of KUKA lightweight robot with high precision position and torque sensors and can be operated in both position control and impedance control modes. In addition, the robot arm also has joint toque sensors at every joint which enable the user to calculate the external force and torque at the end-effector point.

KUKA Lbr iiwa 14 R820 system consists of four main parts (Figure 12). The manipulator is about 1.3 m long and 29.9 kg heavy, has 7 revolute joints driven by brushless motors. The controller operates with KUKA Sunrise.OS which separates the operator control and programming of the robot system. The SmartPAD allows simple manual movements, starting robot applications, activating the safety configuration, jogging, teaching frames, polling inputs. Robot applications are programmed in Java language with KUKA Sunrise.Workbench software (Figure 13) in a development computer.

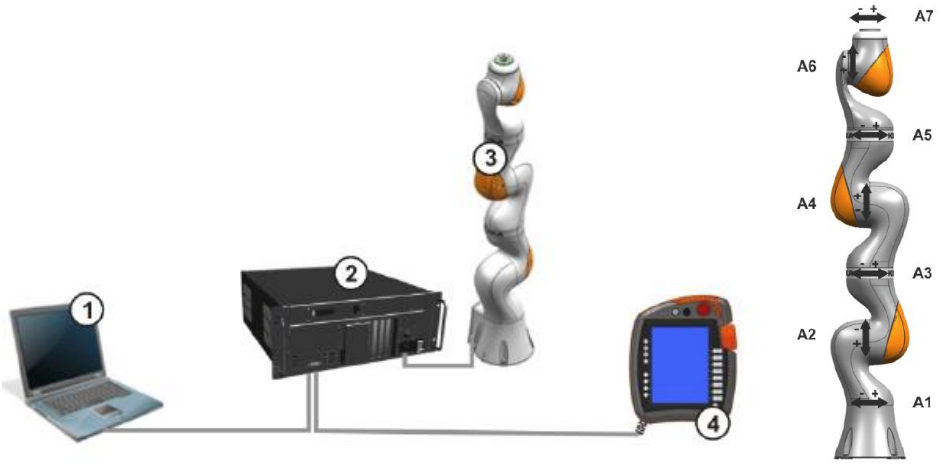


Figure 12: KUKA Lbr iiwa system: 1 - Development computer; 2 - KUKA Sunrise Cabinet robot controller; 3 - Manipulator; 4 - KUKA smartPAD control panel (Os & Workbench, 2014)

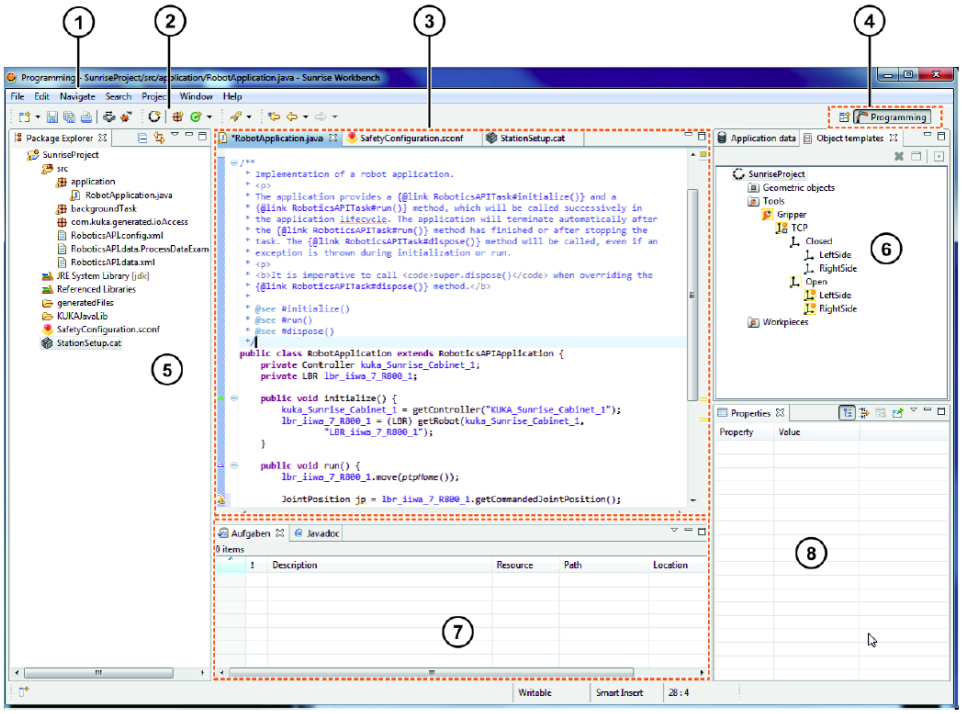


Figure 13: User interface Sunrise.Workbench. 1-Menu bar; 2-Toolbars; 3-Editer area; 4-Perspective selection; 5-Package Explorer view; 6-Application data and Object template views; 7-Tasks and Javadoc views; 8-Properties view (Os & Workbench, 2014)

Designed to work with people, the robot has safety options including joint torque limit, which is used for safety of the project. The joint torque safety interface permits monitoring joint torques to ensure the external torque will not exceed the limit that may cause injury to the person. The main force the robot can exert is the twist about the Z axis. Therefore, the external torque safety is limited to 25Nm. With this value, in the worst scenario, the robot can only exert a force F = T/L = 25/0.571 = 43.78 N to the person (T: torque limit, L: length of the object). In addition, the maximum end-effector velocity is limited to 1500 mm/s in this project. For physical gripping the object, a simple coupler which helps the robot grip an end of the bar is designed. The coupler uses magnets so that it can easily detach from the robot by a vertical force (Figure 14)

Figure 14: Coupler to connector board to robotic end effector

3.2 Impedance control of robotic arm

KUKA LBR iiwa has a KUKA Sunrise.Connectivity SmartServo interface which allows to set new end points of the robot's end effector cyclically during the runtime of the robot motion (Figure 16). Using this interface, the robot's end effector can be programed to move to new positions every a specific period of time. This is appropriate for implementing impedance control. The positions of the end effector will be updated at frequency of 100 Hz. In this project, the robotic arm is constrained to operate in the horizontal plane (Figure 15).

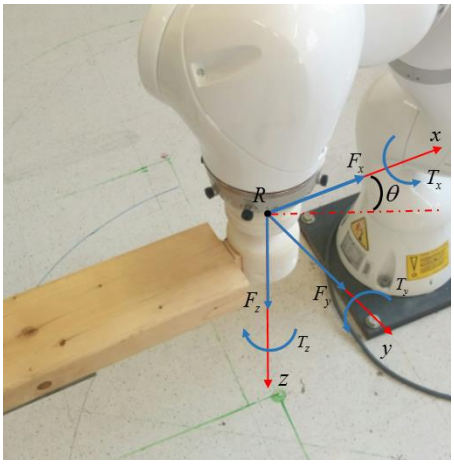


Figure 15: Coordinate system for impedance control

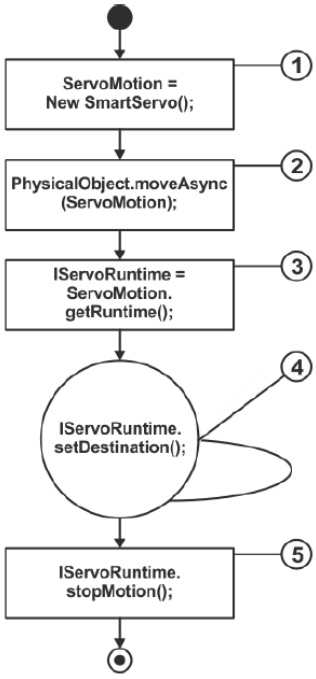


Figure 16: Servo motion in application context. 1-create a new smart servo motion; 2-activate the servo motion; 3-get the run time; 4-set new end points cyclically; 5-end the servo motion (Kuka Sunrise Connectivity, 2014)

A R_{xyz} coordinate system is attached to the end-effector of the robot so that: the X axis is parallel to the object, R is the robot's gripping point. The robot can provide the forces F_x , F_y and torque T_z around the vertical axis which is perpendicular to the xy plane. These values are also described in the end-effector coordinate system R_{xyz} . For compliant interaction, the impedance parameters are assigned respective to the end-effector frame (R_{xyz}), not in the fixed absolute frame (B_{xyz}) (Figure 17). So that these parameters are independent of the position of the end effector of the robot.

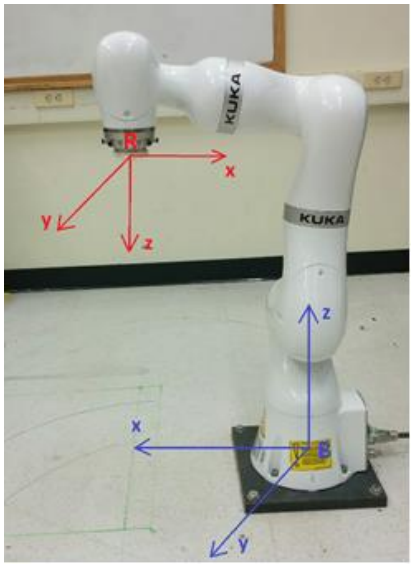


Figure 17: Coordinate systems on the robot

A impedance control law for the movement of the end effector of the robot with mass and viscous friction behavior will be set as follows (Arai et al., 2000):

$$\begin{aligned} F_x &= m_x\ddot{x} + b_x\dot{x} \\ F_y &= m_y\ddot{y} + b_y\dot{y} \\ T_z &= i\ddot{\theta} + c\dot{\theta} \end{aligned} \tag{15}$$

where m_x, b_x, m_y, b_y are mass and friction coefficients in x axis and y axis, i and c are moment inertia and friction coefficient around R. Although the spring factor (stiffness) is usually included in impedance control model, here it is omitted because the effect of the spring factor makes the manipulation task difficult to execute (Ikeura & Inooka, 1995). The acceleration of robot's end effector is:

$$\begin{aligned} \ddot{x} &= (F_x - b_x\dot{x}) / m_x \\ \ddot{y} &= (F_y - b_y\dot{y}) / m_y \\ \ddot{\theta} &= (T_z - c\dot{\theta}) / i \end{aligned} \tag{16}$$

The values of these coefficients are chosen to be as small as possible so that the object can move freely. The external force and torque at the end-effector of the robot are not measured directly. They are calculated through from the measured torques at each joint resulting in some errors and noise. Considering these errors, the impedance parameters cannot be chosen too small in order to keep the system stable (Table 2).

Parameters	Value
$m_x(kg)$	0.5
$b_x(kg/s)$	2.0
$m_y(kg)$	0.5
$b_y(kg/s)$	2.0
$i(kg.m^2)$	0.08
$c(kg.m^2/s)$	0.5

Table 2: Impedance parameters

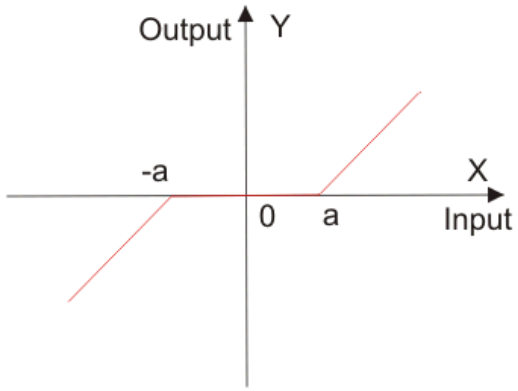


Figure 18: Dead-zone function

To distinguish the error due to measurement and calculation versus force exerted by the person, a dead zone function is used before the forces and torques are fed into control Equation (16). The value of a is chosen to be equal to 3 for F_x and F_y (Figure 18) because without applying external force into the robot

these values range about from -3 to 3 N. As the torque T_z directly depends on measured torque at joint A7 of the robot (as shown in Figure 12), this value is more accurate. Therefore, a dead zone function is not needed for T_z. The position reference of the end effector of the robot is obtained by integrating Equation (16) with a sample time of 10 milliseconds using Euler method.

$$\begin{aligned} \ddot{x}_n &= (F_x - b_x \dot{x}_{n-1}) / m_x \\ \dot{x}_n &= \dot{x}_{n-1} + \ddot{x}_n \Delta t \\ x_n &= x_{n-1} + \dot{x}_n \Delta t \end{aligned} \tag{17}$$

$$\begin{aligned} \ddot{y}_n &= (F_y - b_y \dot{y}_{n-1}) / m_y \\ \dot{y}_n &= \dot{y}_{n-1} + \ddot{y}_n \Delta t \\ y_n &= y_{n-1} + \dot{y}_n \Delta t \end{aligned} \tag{18}$$

$$\begin{aligned} \ddot{\theta}_n &= (T_z - c\dot{\theta}_{n-1}) / i \\ \dot{\theta}_n &= \dot{\theta}_{n-1} + \ddot{\theta}_n \Delta t \\ \theta_n &= \theta_{n-1} + \dot{\theta}_n \Delta t \end{aligned} \tag{19}$$

According to Equation (16) the translation is controlled through F_y, and the rotation is controlled through T_z. In translation mode, the rotation is compressed by setting $T_z = 0$; and in rotation mode, F_y is set equal to zero to avoid translation.

3.3 Using HMMs to for intent recognition

To integrate HMMs in KUKA system which is programmed by JAVA language, a HMMs library Jahmm 0.6.1 – written in Java by Jean-Marc Francois and Willem V. Onsem was adopted. This library implements the Viterbi, Forward-Backward, Baum-Welch and K-Means algorithms for ergodic HMM model. The HMMs program is ran as a background task on the KUKA system. In KUKA system, background tasks are used to perform actions in the background and are parallel to a running robot application. Several

background tasks can run in parallel and independently to each other (Os & Workbench, 2014). There are two types of background tasks:

- Cyclic background tasks which executed cyclically
- And non-cyclic background tasks which executed only once

The cyclic task is used for running online intent recognition task. The HMMs and switching function run in background to provide the recognition results which then are used by the impedance control program running as the main robot application.

In this project, ergodic HMM models are chosen. Figure 19 shows an example of an ergodic three state HMM model, in which all transitions between each states are non-zero. The observations are continuous signal. Although it is possible to quantize the signals for the use of discrete HMMs, it would be advantageous to use HMMs with continuous observation densities (Rabiner, 1989). The data inputs are the torque in z axis (T_z), torque in y axis (T_y) and the force in x - axis F_x (Figure 20). HMMs with multivariate Gaussian observation are used with each observation is a vector $\mathbf{x}$:

$$\mathbf{x} = \begin{bmatrix} T_y \\ T_z \\ F_x \end{bmatrix} \tag{20}$$

The frequency of collecting data which is also the frequency of running HMMs for online intent recognition is 25 Hz. This frequency is high in comparison to human hand's motion's frequency which is up to 5 Hz (Flanagan & Johansson, 2002) (Samur, 2012). The combination of a series of observation vector over time creates an observation sequence.

$$O = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\} \tag{21}$$

The observation output probability density matrix $B = \{b_j(\mathbf{x})\}$ is form by multivariate Gaussian density $b_j(\mathbf{x})$ (with $j = 1, 2..., N$):

$$b_j(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^k \,|\, \boldsymbol{\Sigma}_j \,|}} \exp\left(-\frac{1}{2}\left(\mathbf{x} - \boldsymbol{\mu}_j\right)^T \boldsymbol{\Sigma}^{-1}\left(\mathbf{x} - \boldsymbol{\mu}_j\right)\right) \tag{22}$$

where

$\boldsymbol{\mu}_j \in R^k$: mean vector of probability density of state j^{th}

$\boldsymbol{\Sigma}_j \in R^{k \times k}$: covariance matrix of probability density of state j^{th}

k : dimension of observation vector $\mathbf{x}$ ($k = 3$ in this case)

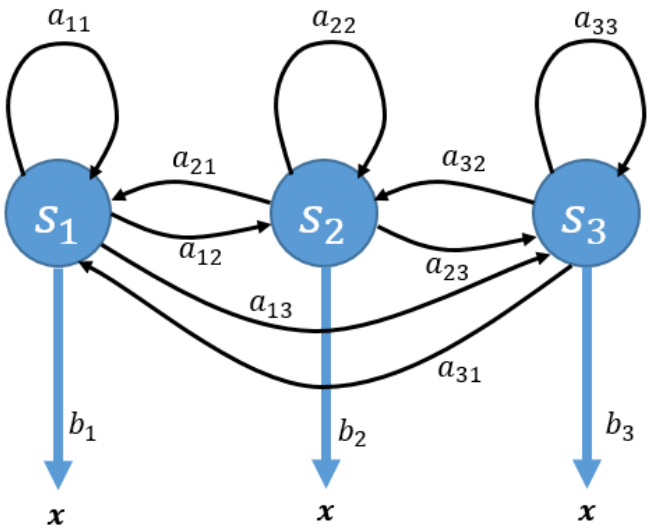


Figure 19: Example of an ergodic three state HMM model

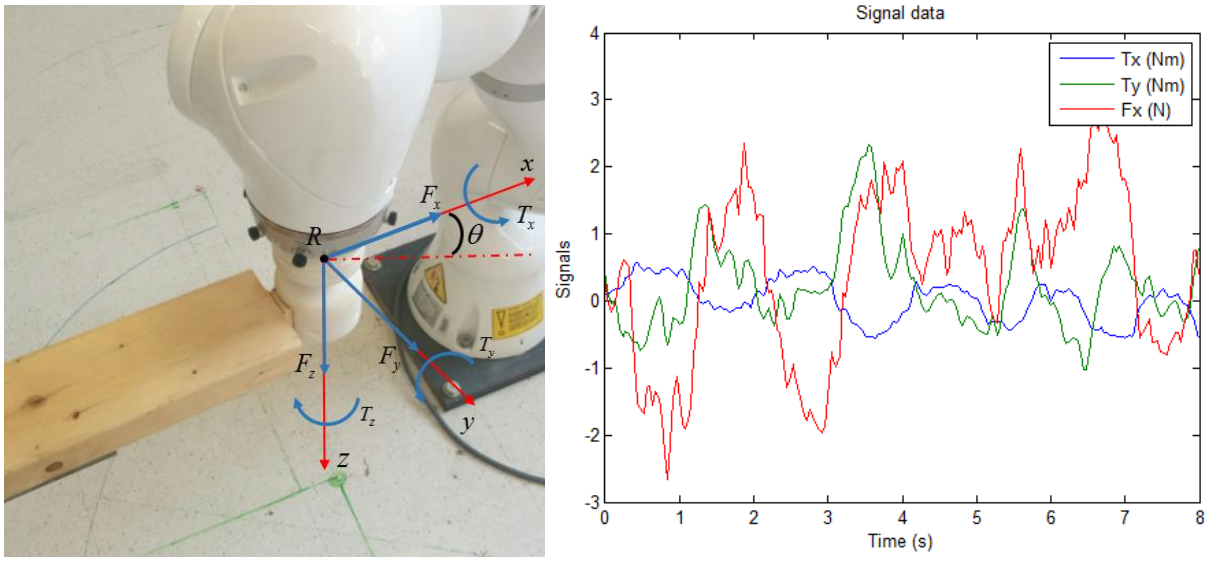


Figure 20: Force and torque signals at the robot's end-effector

The initialization for HMMs parameters before training are found using K-mean cluster algorithm. First, the K-mean algorithm with the centroid distance will separate all observation vectors in the training data set into N data clusters (N is the number of states in HMMs). The observation output probability densities' ($Opdf$) parameters are found by fitting the data clusters. For example, mean and covariance of $Opdf_j$ are found by fitting the data cluster j^{th}. To initialize the state transition probability matrix $A=\{a_{ij}\}$, all observation vectors in sequences are classified into each data clusters and given the state number according to their clusters they are belong to. Then numbers of transitions from the state i to the state j are counted. Finally, those numbers are normalize to get the values of a_{ij}. The initial state probability matrix $\pi=\{\pi_i\}$ is found by the similar method. First, all first observation vectors in each sequences of training data set are classified. The state numbers are given for each of first observation vectors according to the data clusters they belong to. The number of states appear as first state in sequences are counted, then are normalized to get π_i.

3.3.1 Using two HMMs with data collected in only impedance control

During manipulation, the person has two basic intents: rotation or translation. Therefore, a HMM for each of the intent modes is needed: λ_R for rotation and λ_T for translation. Figure 21 shows the scheme of the intent recognition which includes feature extraction and intent classification.

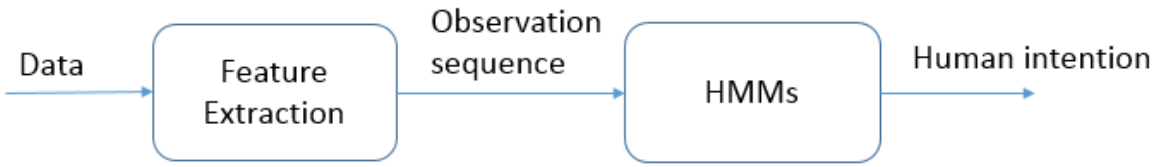


Figure 21: Intent recognition scheme

For collecting data for training the HMMs offline, the robot is controlled with relatively low impedance control in horizontal plane and a high stiffness is set in vertical direction so that the robot can leverage the mass of the object. The data of force F_x, F_y, F_z and torque T_x, T_y, T_z at the gripping point of the robot (Figure 22) is collected at sample period of 40 milliseconds for each intent mode.

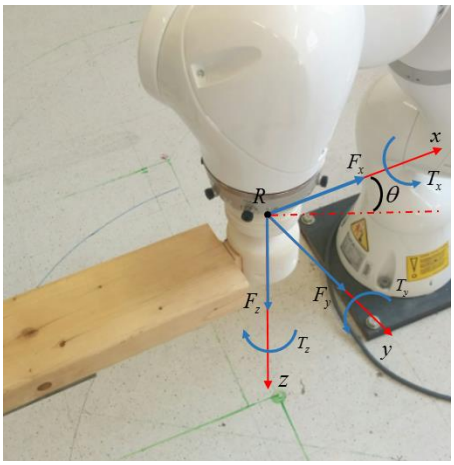


Figure 22: Coordinate system at the robot's gripping point (R)

Although it is common that the data is normalized before feeding into HMMs, in this project the force and torque data is not normalized because the input signals' ranges are relatively the same. For training the HMMs, 90 data sequences were collected for each intent mode (Figure 23). 60 data sets are used for training the HMMs and 30 data sets are used for evaluating the models.

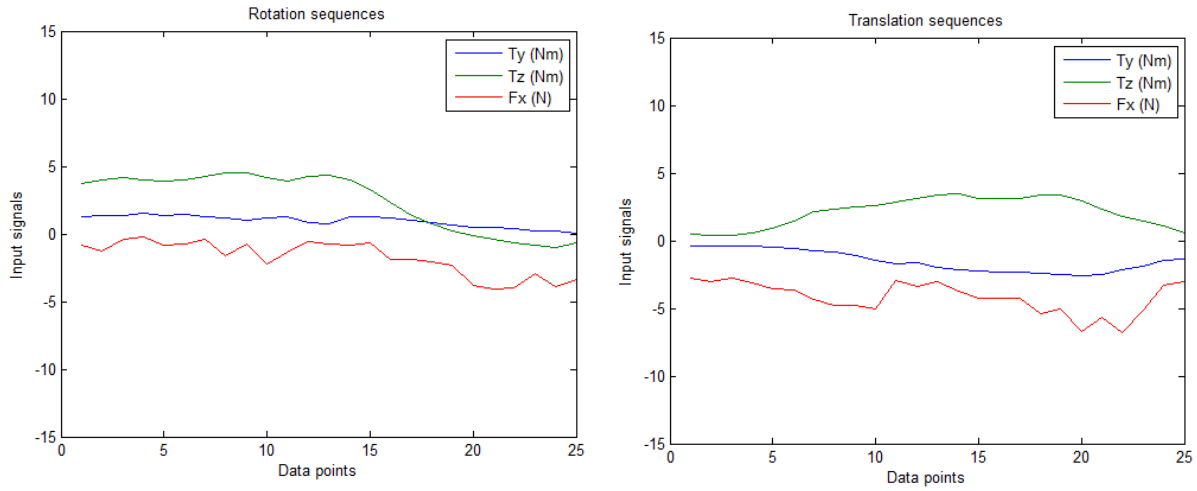


Figure 23: Examples of translation and rotation sequences

The length of sequences will be the number of data points in the sequences. For number of states, in general, there is no theoretical method for determining the number of states of HMMs (Rabiner, 1989). One may think of Bayesian information criterion (BIC). However, BIC only works well for optimizing a single HMMs with respect to computational effort, it is not in the case several HMMs work together (Nikolay Stefanov, Passenberg, Peer, & Buss, 2013). Therefore, in this project, the number of states of HMMs is considered through the accuracies on testing data sets.

For online recognition, the observation sequences are extracted from real-time data. That means a new observation sequence is formed at each sample time of 40 milliseconds (frequency of 25 Hz) and then passed into the trained HMMs. This sample time is high in comparison to human hand's motion's frequency which is up to 5Hz (Flanagan & Johansson, 2002) (Samur, 2012). The HMM model with higher likelihood will define the intent mode of the subject. The output of the problem 1 will be the likelihood

that model $\lambda = (A, B, \pi)$ generates the observation sequence $O = \{o_1, o_2, ..., o_T\}$. Because this value could be very small, it is more convenient to use logarithmic scale $LL(O|\lambda)$.

$$LL(O|\lambda) = \ln(p(O|\lambda)) \tag{23}$$

That means the translation mode is true if $LL(O, \lambda_T) > LL(O, \lambda_R)$. The similar holds for rotation mode.

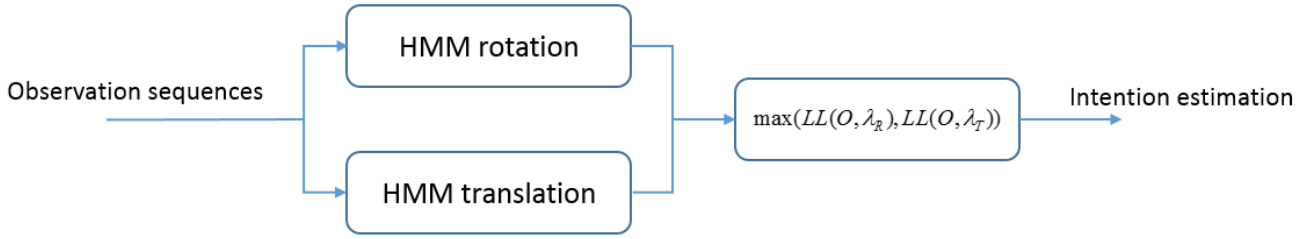


Figure 24: Intent estimation using two HMMs

3.3.2 Using two HMMs and four HMMs with data collected in separate modes

When using two HMMs as mentioned above, the training data is collected when the robot ran only in impedance control and without switching modes. Later the trained HMMs are used for recognizing the person's intent then switching mode according to the intent. Because the training data and working data are at different conditions, the performance may decrease. This problem can be solved by collecting data in separate modes. First, a person intents do rotation and translation actions while the robot is running in rotation mode. The data for translation intent and rotation intent is collected. Latter, the person intents do rotation and translation actions while the robot is running in translation mode. The data for translation intent and rotation intent is collected. 90 sequences are extracted for each intent in while the robot is running in rotation mode. Similarly, 90 sequences are extracted for each intent in while the robot is running in translation mode. Two-thirds of these sequences are used for training and one third are used for testing to determine the appropriate HMM model. The observation sequence is extracted online, then

passed into the HMMs. The data is used for trained two HMMs (λ_R and λ_T). The intent recognition result is determined by comparison of the two likelihoods (Figure 24).

Because of separate modes, two HMMs in rotation mode (λ_{T_inR} and λ_{R_inR}) can be used when the robot is in rotation mode, two HMMs in translation mode (λ_{T_inT} and λ_{R_inT}) can be used when the robot is in translation mode. In this case, the training data for two HMMs in rotation mode are collected as the person tries to do rotation and translation actions while the robot is running in rotation mode. Similarly, the training data for two HMMs in translation mode are collected as the person tries to do rotation and translation actions while the robot is running in translation mode.

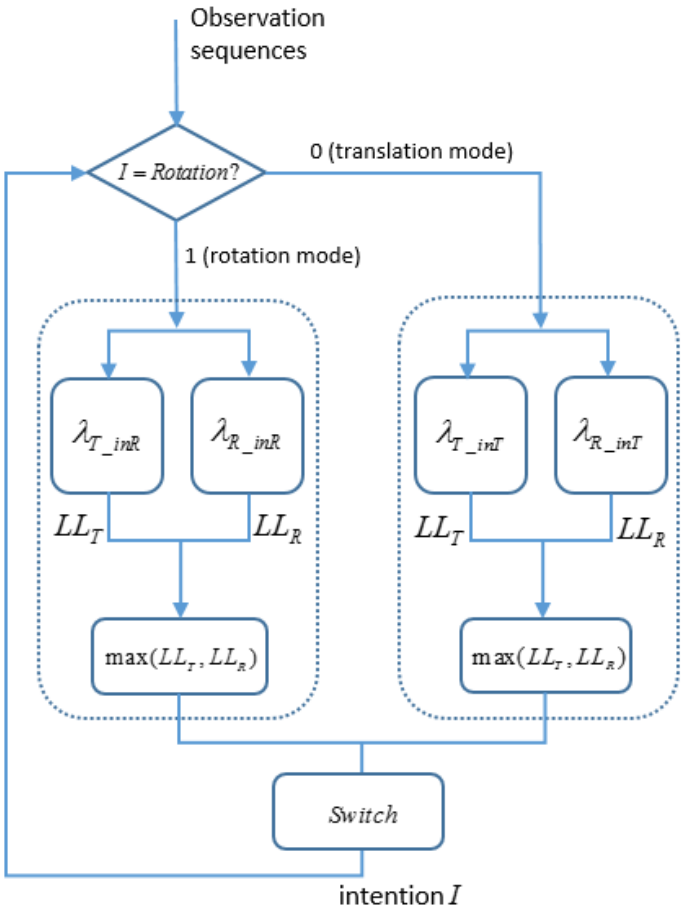


Figure 25: Intent estimation diagram using four HMMs

The observation sequence is extracted online, then passed into the HMMs in rotation mode (λ_{T_inR} and λ_{R_inR}) if the current mode is rotation. Likewise, the sequence is passed into the HMMs in translation mode (λ_{T_inT} and λ_{R_inT}) if the current mode is translation. The intent recognition results are determined by comparison of the two likelihoods. With this approach, chattering can happen when the two HMMs in rotation (λ_{T_inR} and λ_{R_inR}) give the result as translation intent, and the in next time step two HMMs in translation (λ_{T_inT} and λ_{R_inT}) give the result as rotation intent. This is solved by a switch block in which the system only switches when there is an agreement in results between the two pairs of HMMs. For example, the system is currently in rotation mode, two HMMs in rotation will do the recognition and give the result as translation intent. That means the next data sequence will be fed into two HMMs in translation. The system only switches into translation if the two HMMs in translation also give the results as translation (Figure 26). For better explanation, the abbreviation for these methods are described in Table 3.

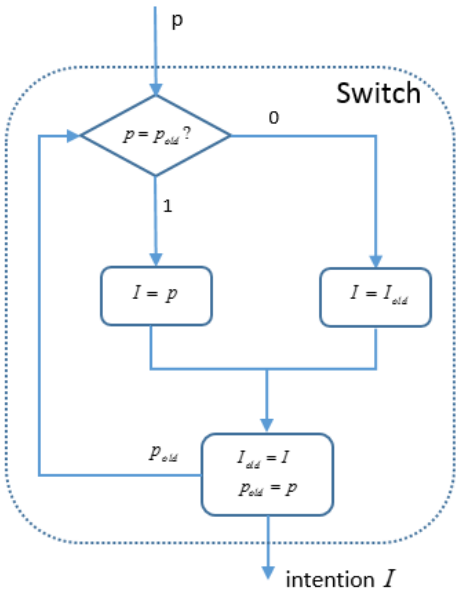


Figure 26: Switch block

Abbreviation	Meaning
2HMMs_OnlyImp	Two HMMs with data collected in only impedance control.
2HMMs_RotTra	Two HMMs with data collected in separated modes.
4HMMs	Four HMMs with data collected in separated modes.
SwitchFunc	Switching function based on force magnitude.

Table 3: Abbreviation of different methods

3.4 Switching function based on force magnitude

In (Karayiannidis et al., 2014), authors proposed a method to map human's intent of rotation versus translation based on force magnitude F. For movements in horizontal plane, force magnitude is calculated:

$$F = \sqrt{f_x^2 + f_y^2} \tag{24}$$

The rotation is defined when force magnitude is under a threshold, and translation is when the magnitude is beyond the threshold. The intuition of this idea is that the person is likely to reduce the forces to allow a switch from translation to rotation mode; and while in rotation mode, the person must increase the force to switch to translltion mode. A hysterisis δ_f is introduced to avoid unintended switching chattering due to inaccurate operation or noisy measurements (Figure 27). Although in (Karayiannidis et al., 2014), the values of threshold f_0 and hysteresis δ_f are determined empirically, in this project, these values are determined through the mean and deviation of recorded interaction force's magnitude so that the switching funtion can give a fair rotation and translation region.

$$\delta_f = \sigma \tag{25}$$

$$f_0 = \mu - \frac{\sigma}{2} \tag{26}$$

Where μ is the mean and σ is the deviation.

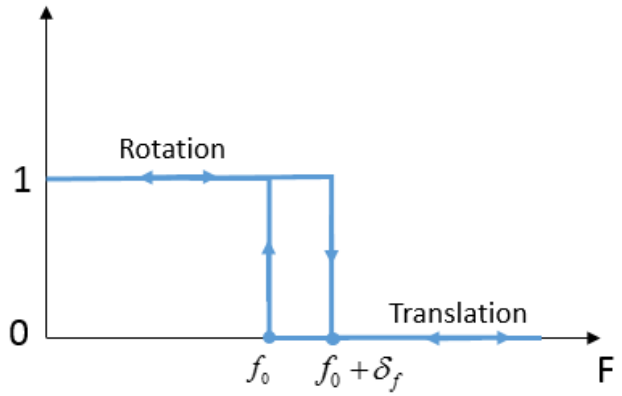


Figure 27: Switching function based on force magnitude

3.5 Experimental protocol

In the experiment, a weighted wood board (571 x 89 x 50 millimeters, 2.0 kg) is chosen as the object for the collaborative manipulation tasks (Figure 28). A male person (178cm, 70 kg) is asked to hold the end of the object with one hand as they find comfortable. This prevents the person from applying a large torque at the gripping point. For collecting training data for HMMs, the person was asked to do rotation and translation movements. The interaction forces and torques were recorded and then separated into the sequences as described in section 3.3.

To evaluate the method, the person was asked to do some trials first and then move the object follow task TRANSLATION - a straight line with translation motion, ROTATION - circular with rotation motion, and COMBINED - translation followed by rotation and vice versus. In the TRANSLATION task, the person tried to translate the object from a starting position to another position. Because the robot's position is fixed, and the robot's also try to keep the height of object fixed in manipulation tasks, the distance of translation

is limited. In this case, the straight line for translation is 60 cm length. In the ROTATION task, the person rotates the object around the robot's gripping point 180°. In the COMBINED task, the path is a combination of a straight line of 60 cm length and a circular of 135°. The person was ask to do translation movement following the straight line and rotation movement following the circular. The switching between rotation and translation is continuous without stopping. Moving following the paths, they can walk following the object which is similar in real case of collaborative manipulations. For each case, the person held the object with their right hand and moved the object in one direction. After the path is finished, the person stops for one to three seconds and switches into the opposite direction by holding the object by their left hand and moving following the paths (Figure 28). In each direction, the person moved the object 5 times.

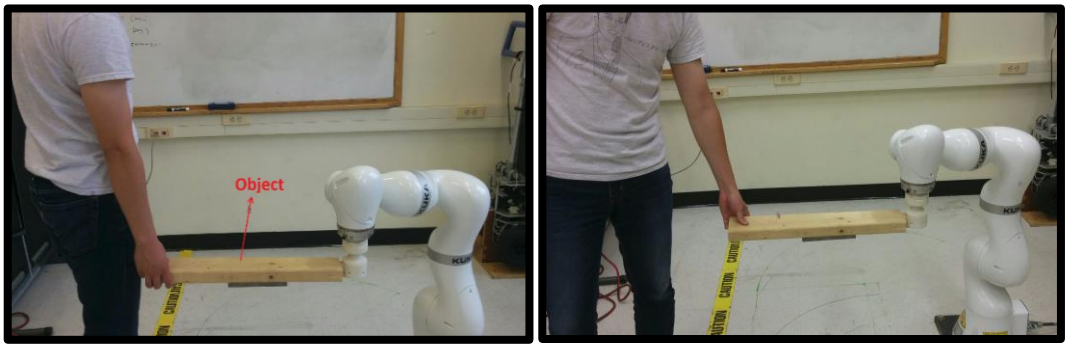


Figure 28: Human and robot jointly hold the object when moving in two directions

One problem of recognizing using switching function is that the system cannot correctly detect when fast rotation or slow translation is desired. For the comparison, testing with fast rotation and slow translation is conducted. The person is asked to do the ROTATION task – circular with rotation at a fast speed (that means the person need to apply a larger force to the object), and do the TRANSLATION task – straight line with translation in slow speed (that means the person need to apply less force to the object). The robot is set to rotation mode during ROTATION task and translation mode during TRANSLATION task so that the predictions of all methods (HMMs and switching function) are compared.

During collaborative tasks, the number of switches is important. A reduced number of switches will result in a task done more smoothly. Therefore, the number of switches between two modes is counted in each trial. In the TRANSLATION and ROTATION tasks the number of switches is ideally zero because the person only intends to do translation or rotation. In the COMBINED task in which both rotation and translation are involved the number of switches is ideally one. In addition, the accuracy of the prediction is calculated as the ratio between number of correct predictions $n_{correct}$ and total number of predictions N . This value can also described by the correct prediction time $t_{correct}$ divided by the total executed time T as the prediction frequency is fixed.

$$Accuracy = \frac{n_{correct}}{N} \times 100\% = \frac{t_{correct}}{T} \times 100\% \tag{27}$$

CHAPTER 4

RESULTS

The experiment is set up as described in section 3.5. For each method, ROTATION task is done first, then followed by TRANSLATION and COMBINED tasks. Figure 29 – Figure 32 show examples of each task. Figure 29 shows an example of the TRANSLATION task using four HMMs method (4HMMs) in which the object is moved with intent to translate. The prediction is indicated by values: 0 – translation, 1 – rotation. At the beginning, due to the delay of prediction, there was an error which caused the object to rotate a little. The system latter detected the person's intent as translation. At very end, the person tried to rotate the object back to make it align with the desired stopping position. That makes the robot switch into rotation mode. The number of mode switches is twice in this case. Figure 30 shows an example of circular path following with rotation (task R) in which the object is rotated 180^0. As the system predicted correctly in this case, the object rotated and the robot's gripping point did not move. For the COMBINED task, Figure 31 shows the object was translated then rotated about the robot gripping point. At the beginning, there was a small error due to delay of the system in prediction, then the translation intent was predicted. At the end of the straight line segment, the person switched into rotation.

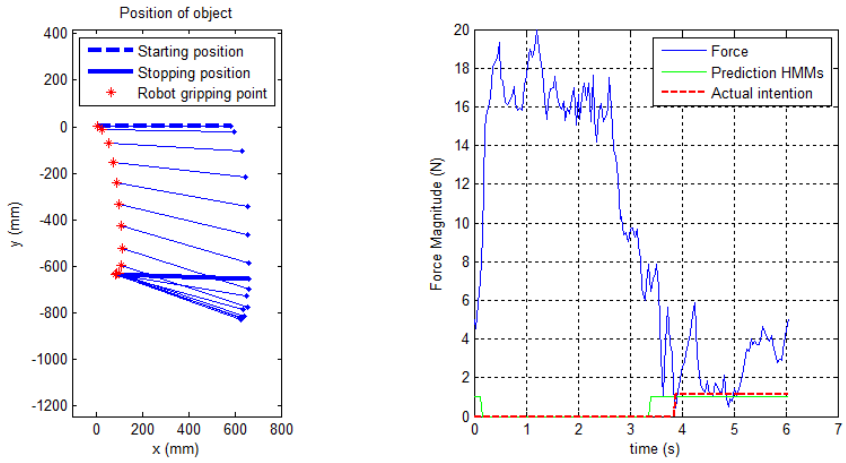


Figure 29: Example of translation movement using 4HMMs

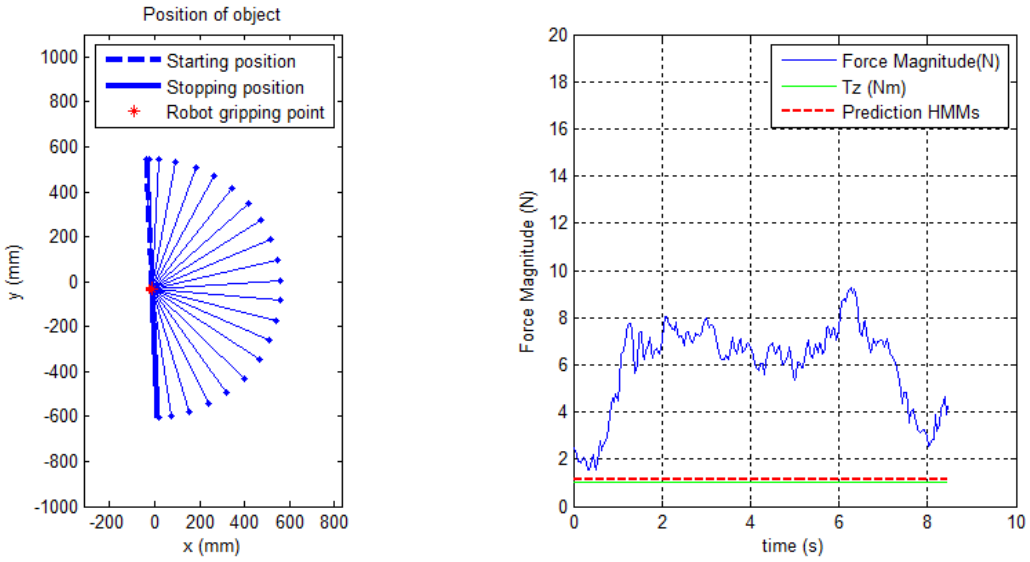


Figure 30: Example of rotation movement using 4HMMs

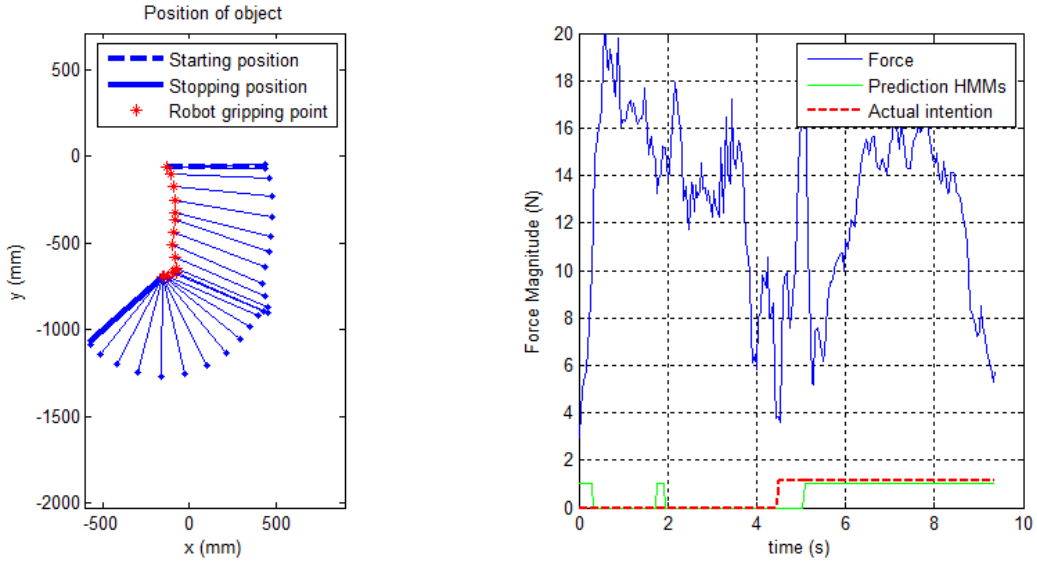


Figure 31: Example of translation followed by rotation using 4HMMs

4.1 Using two HMMs with data in only impedance control (2HMMs_OnlyImp)

In order to determine the appropriate model for HMMs, different models with different number of states are considered. Table 4 shows the accuracies on the testing data sets when using HMMs with different number of states (2, 3, 4, 5 states) and different sequence lengths (10, 15, 20, 25, 30, 35 data

points). As the accuracy of model with two states and length of 25 points is fairly high with least number of states, this model is used in this case. Table 5 - Table 7 show the data for 10 trials for ROTATION, TRANSLATION and COMBINED tasks. The abbreviation "No. of switching" means number of switching, "Avg." means average values. The accuracy of task ROTATION is 84.7%, task TRANSLATION is 78.1%, task COMBINED is 73.1% and overall accuracy is 78.7%.

No. of States\length	**10**	**15**	**20**	**25**	**30**	**35**
2	88.3	90.0	95.0	**96.7**	96.7	96.7
3	90.0	93.3	93.3	95.0	95.0	95.0
4	88.3	88.3	95.0	95.0	95.0	95.0
5	86.6	91.6	96.7	95.0	95.0	95.0

Table 4: Accuracy of predictions with different number of states and sequence lengths

Evaluation\Trial	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg.**
No. of switching	2	1	4	5	5	2	1	0	3	2	2.50
Run time (second)	8.20	8.32	9.72	9.88	8.84	8.12	8.04	8.00	8.80	9.52	8.74
Duration of Prediction Error (second)	0.24	0.92	2.520	3.680	1.320	0.88	0.92	0	1.84	1.60	1.39

Table 5: Results task ROTATION using 2HMMs_OnlyImp

Evaluation\Trial	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg.**
No. of switching	10	3	4	2	6	2	6	4	6	2	4.50
Run time (second)	11.08	6.04	8.40	6.40	10.40	6.64	9.00	7.44	8.16	9.20	8.28
Duration of Prediction Error (second)	3.48	0.96	1.48	1.20	3.32	1.72	2.32	1.00	1.44	1.88	1.88

Table 6: Results task TRANSLATION using 2HMMs_OnlyImp

Evaluation\Trial	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg.**
No. of switching	2	4	6	11	6	16	4	10	7	10	7.6
Run time (second)	7.64	10.36	8.88	16.32	8.76	16.68	7.88	20.16	8.84	13.12	11.86
Duration of Prediction Error (second)	0.40	3.48	2.36	7.72	2.52	4.72	1.08	7.40	3.00	1.96	3.46

Table 7: Results task COMBINED using 2HMM_OnlyImp

4.2 Using two HMMs with data collected in separated modes (2HMMs_RotTra)

First, the number of states in HMM models is determined via the accuracies on the testing data set. As 5 state HMMs has highest accuracy (Table 8), this model is used. Table 9, Table 10, Table 11 show the data of ten trials for task ROTATION, TRANSLATION, and COMBINED using HMMs with data collected in rotation and translation mode. The accuracy of test ROTATION is 92.9 %, test TRANSLATION is 81.6 %, and test COMBINED is 88.0%, and 87.5% for all 3 tests.

Number of states	Accuracy (%)
2	93.3
3	93.3
4	93.3
5	**95.8**
6	95.0
7	95.0

Table 8: Accuracy when using HMMs with different number of states

Evaluation\Trial	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg.**
No. of switching	3	3	1	2	1	2	1	2	3	0	1.8
Run Time (second)	9.44	9.68	7.68	7.84	8.24	8.16	7.76	8.00	8.72	7.48	8.3
Duration of Prediction Error (second)	1.72	0.44	0.44	0.04	0.96	0.68	0.28	0.44	1.16	0	0.62

Table 9: Results of task ROTATION with 2HMMs_RotTra

Evaluation\Trial	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg.**
No. of switching	1	2	6	3	4	1	3	5	1	3	2.9
Run time (second)	6.28	6.60	7.32	8.40	7.80	6.64	6.92	9.04	6.44	8.60	7.40
Duration of Prediction Error (second)	0.08	1.08	1.92	1.60	2.44	0.72	1.20	2.84	0.76	1.56	1.42

Table 10: Results of task TRANSLATION with 2HMMs_RotTra

Evaluation\Trial	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg.**
No. of switching	1	2	3	4	4	4	4	8	5	6	4.1
Run time (second)	7.44	11.04	6.92	10.60	7.96	11.76	8.12	13.60	7.76	16.68	10.18
Duration of Prediction Error (second)	0.28	0.76	0.80	0.96	1.64	1.76	1.44	1.40	1.08	1.92	1.20

Table 11: Results of task COMBINED with 2HMMs_RotTra

4.3 Using four HMMs with data collected in separated modes (4HMMs)

First, the number of states for HMMs is determined through the accuracies on the extracted testing data set. Two state and three state HMMs in rotation mode and 3 state HMMs in translation mode have the highest accuracy (Table 12). For simplification, three state models are used for all HMMs. The accuracy of test ROTATION is 92.0 %; test TRANSLATION is 80.6 % and test COMBINED is 88.1%, and 86.9% for all 3 tests.

No. of States\Cases	HMMs in Rotation mode	HMMs in Translation mode
2	**98.3**	93.3
3	**98.3**	**96.7**
4	95.0	95.0
5	95.0	95.0

Table 12: Accuracy when using HMMs with different number of states

Evaluation\Trial	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg.**
No. of switching	0	2	0	4	1	1	2	0	0	2	1.2
Run time (second)	7.96	10.04	8.24	10.28	8.80	9.52	11.32	8.72	8.48	9.16	9.25
Duration of Prediction Error (second)	0	1.92	0	2.04	0.2	2.72	0	0	0	0.96	0.78

Table 13: Results task ROTATION using 4HMMs

Evaluation\Trial	1	2	3	4	5	6	7	8	9	10	Avg.
No. of switching	2	2	2	2	4	2	2	2	4	1	2.3
Run time (second)	6.04	7.04	10.84	7.64	7.20	8.08	7.88	8.00	7.92	7.08	7.77
Duration of Prediction Error (second)	0.36	1.36	4.76	1.6	1.04	1.8	1.16	1.48	1.8	0.76	1.61

Table 14: Results task TRANSLATION using 4HMMs

Evaluation\Trial	1	2	3	4	5	6	7	8	9	10	Avg.
No. of switching	2	4	3	3	4	4	4	2	3	1	3.0
Run time (second)	7.84	12.56	9.4	12.16	8.6	12.52	9.24	11.56	8.44	8.36	10.07
Duration of Prediction Error (second)	0.40	2.84	1.76	1.64	1.84	1.84	0.68	0.80	0.44	0.32	1.26

Table 15: Results task COMBINED using 4HMMs

4.4 Switching function based on force magnitude

The switching function's parameters are determized through the mean and deviation of interaction force magnitudes. As the mean of interaction force magnitudes is $\mu = 9.13$ and deviation is $\sigma = 5.33$, the hysteresis is chosen to be equal to the deviation: $\delta_f = 5.33$, and the threshold $f_0 = 6.47$ (Equation 25, 26)

Evaluation\Trial	1	2	3	4	5	6	7	8	9	10	Avg.
No. of switching	0	0	0	0	0	0	0	0	0	0	0
Run time (second)	12.36	10.60	12.12	11.36	11.8	11.04	11.80	11.16	12.08	11.04	11.54
Duration of Prediction Error (second)	0	0	0	0	0	0	0	0	0	0	0

Table 16: Results task ROTATION using SwitchFunc

Evaluation\Trial	1	2	3	4	5	6	7	8	9	10	Avg.
No. of switching	2	2	2	2	2	2	2	2	2	2	2
Run time (second)	6.80	7.36	6.44	7.16	5.80	6.72	5.32	5.80	5.32	5.68	6.24
Duration of Prediction Error (second)	0.68	0.60	0.56	1.00	0.72	1.04	0.64	0.72	0.64	0.72	0.73

Table 17: Results task TRANSLATION using SwitchFunc

Evaluation\Trial	1	2	3	4	5	6	7	8	9	10	Avg.
No. of switching	2	2	4	2	2	2	2	2	2	2	2.2
Run time (second)	8.84	12.92	9.52	10.76	8.40	11.88	7.92	11.28	7.88	10.32	9.93
Duration of Prediction Error (second)	0.36	0.52	1.32	0.60	0.48	0.36	0.64	0.52	0.52	0.76	0.68

Table 18: Results task COMBINED using SwitchFunc

In test ROTATION (Table 16), it is understandable that the number of switching is zero. As the person try to do rotation slowly, the predictions are all correct as rotation. In test TRANSLATION, at the beginning of each trial, the robot always stays in rotation mode as the force magnitude is small. Therefore, when the person exerts a force to move the object, the force magnitude increases to reach the threshold and leads to a switch from rotation to translation. At the end of each trial, the person decreases the applied force that causes another switch from translation to rotation. Therefore the number of mode switches is two (Table 17). In test COMBINED which involved both translation and rotation actions, the number of switches slightly increases (Table 18). The reason is that during the translation segment, although the person intended to translate the object, it takes a period of time for them to increase the applied force to reach the threshold of the translation region in the switching function. The accuracy for test ROTATION is 100%, for test TRANSLATION is 88.2%, for test COMBINED is 93.7 % and overall accuracy is 93.9%. Table 19 Table 20 shows the summary of number of switches and the accuracy with different methods. The results shows that the numbers of switches significantly decrease with 2HMMs_RotTra and 4HMMs methods in comparison to 2HMMs_OnlyImp method ($p < 0.05$). In addition, the accuracy with 2HMMs_RotTra and 4HMMs methods are significantly better than the accuracy with 2HMMs_OnlyImp method ($p < 0.05$). And the accuracy of SwitchFunc method are significant better than all other methods ($p < 0.05$).

Methods	Number of mode switches							
	ROTATION		TRANSLATION		COMBINED		Overall	
	Mean	Deviation	Mean	Deviation	Mean	Deviation	Mean	Deviation
2HMMs_OnlyImp	2.50	1.72	4.50	2.55	7.60	4.17	4.87	3.59
2HMMs_RotTra	1.80	1.03	2.90	1.73	4.10	1.97	2.93	1.84
4HMMs	1.20	1.32	2.30	0.95	3.00	1.05	2.17	1.31
SwitchFunc	0.00	0.00	2.00	0.00	2.20	0.63	1.40	1.07

Table 19: Number of mode switching with different methods

Methods	Accuracy (%)							
	ROTATION		TRANSLATION		COMBINED		Overall	
	Mean	Deviation	Mean	Deviation	Mean	Deviation	Mean	Deviation
2HMMs_OnlyImp	84.8	10.9	78.1	6.5	73.1	12.5	78.7	11.0
2HMMs_RotTra	92.9	5.8	81.6	9.4	88.0	5.0	87.5	8.2
4HMMs	92.0	10.8	80.6	10.1	88.1	7.2	86.9	10.3
SwitchFunc	100	0	88.2	2.3	93.7	3.1	93.9	5.3

Table 20: Accuracy of predictions with different methods

4.5 Fast rotation and slow translations

The switching function based on force magnitude is proposed in (Karayiannidis et al., 2014). A disadvantage of this approach is that the person has trouble successfully completing a fast rotation or slow translation. In the above tests, the person chose his speed of doing actions by himself. The average speeds of rotation and translation are calculated based on data in ROTATION and TRANSLATION tasks (Table 21). When fast rotation is desired, the force magnitude is above the switching threshold and the controller will infer this as translation mode. Similarly, in slow translation, force magnitude is below the switching threshold so the controller is likely to infer as rotation mode.

Figure 32 shows the cases of fast rotation (left) and slow translation (right). In the fast rotation, most of predictions with the switching function are wrong as translation (values are 0), while predictions with using HMMs are correct as rotation (values are 1). Similarly in the slow translation case, while predictions

based on switching function are wrong as rotation, most of predictions with HMMs are correct as translation. When fast rotations or slow translations are required, the HMMs methods still perform well, while the switching function based on force magnitude perform poorly (Table 23, Table 24). The results shows that HMMs methods perform significantly better than the switching function in fast rotation and slow translation ($p < 0.05$).

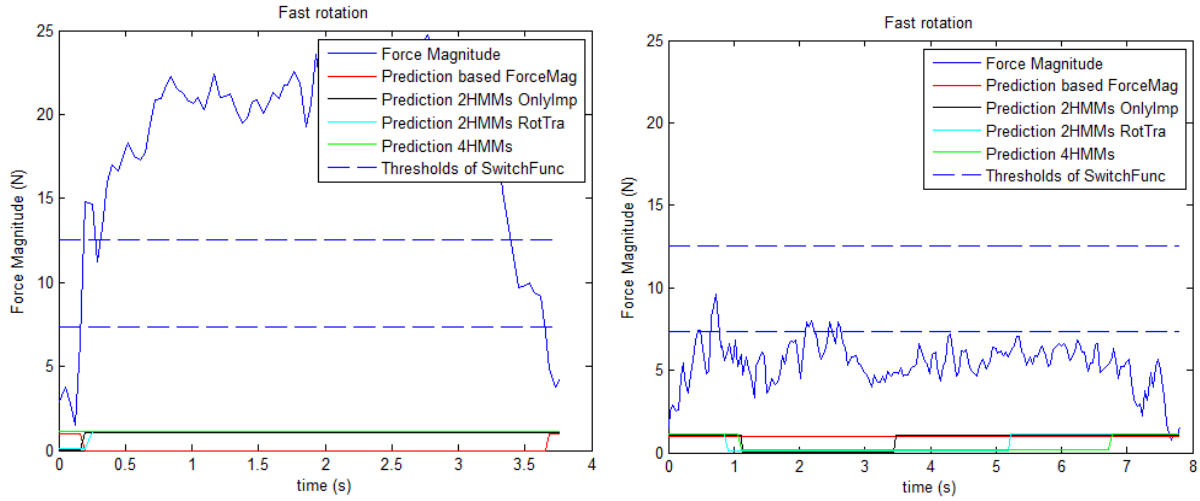


Figure 32: Fast rotation and slow translation

Tasks	**Self-selected speeds**		**Fast Rotation**		**Slow Translations**	
	Mean	Deviation	Mean	Deviation	Mean	Deviation
Rotation (rad/s)	0.34	0.05	0.76	0.04	x	x
Translation (cm/s)	15.83	1.73	x	x	7.42	0.58

Table 21: Self-selected speeds and fast rotation/ slow translation speed

Methods	**Accuracy (%)**					
	Fast Rotation		**Slow Translation**		**Overall**	
	Mean	Deviation	Mean	Deviation	Mean	Deviation
2HMMs_OnlyImp	90.4	12.9	43.3	23.7	67.0	30.4
2HMMs_RotTra	93.7	4.7	62.9	14.5	78.3	19.0

4HMMs	92.9	8.6	69.7	20.1	81.3	19.2
SwitchFunc	10.7	4.0	2.0	4.5	6.4	6.0

Table 22: Accuracies in fast rotation and slow translation tasks

Evaluation\Trial		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg.**
Run time (second)		4.32	4.12	3.92	4.16	3.72	4.16	3.84	4.12	3.88	3.92	4.02
Duration of Prediction Error (second)	**2HMMs_OnlyImp**	0.48	1.52	0.00	1.08	0.00	0.60	0.28	0.00	0.00	0.00	0.40
	2HMMs_RotTra	0.56	0.00	0.32	0.00	0.36	0.20	0.32	0.00	0.36	0.36	0.25
	4HMMs	0.00	0.48	0.00	0.24	0.00	0.96	0.00	0.52	0.00	0.68	0.29
	SwitchFunc	3.40	3.84	3.52	3.72	3.40	3.76	3.48	3.60	3.52	3.60	3.58

Table 23: Results when fast rotation is required

Evaluation\Trial		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg.**
Run time (second)		7.00	9.12	7.60	7.80	6.96	7.84	6.72	8.80	6.92	8.12	7.69
Duration of Prediction Error (second)	**2HMMs_OnlyImp**	1.44	6.56	4.28	5.44	3.64	5.20	1.72	8.76	4.52	3.04	4.36
	2HMMs_RotTra	2.24	4.24	4.04	3.48	3.36	1.28	2.40	2.68	3.60	0.96	2.83
	4HMMs	2.56	0.20	3.32	2.16	3.00	0.32	2.16	2.92	4.72	1.00	2.23
	SwitchFunc	7.00	9.12	7.60	7.80	6.24	7.84	6.72	8.80	6.92	7.24	7.53

Table 24: Results when slow translation is required

CHAPTER 5

DISCUSSION

Chapter 4 presented the results of testing with three manipulation tasks using four methods. For comparison, bar graphs of number of switching time and accuracy of different methods: two HMMs with data collected in only impedance control (2HMMs_onlyImp), two HMMs with data collected in different defined intent mode (2HMMs_RotTra), four HMMs with data collected in different defined intent mode (4HMMs), and switching function based on force magnitude (SwitchFunc) are used (Figure 33, Figure 34). The bar graphs show means μ_i with the range of one deviation σ_i ($\mu_i \pm \sigma_i$) for the accuracy and number of switches of each method.

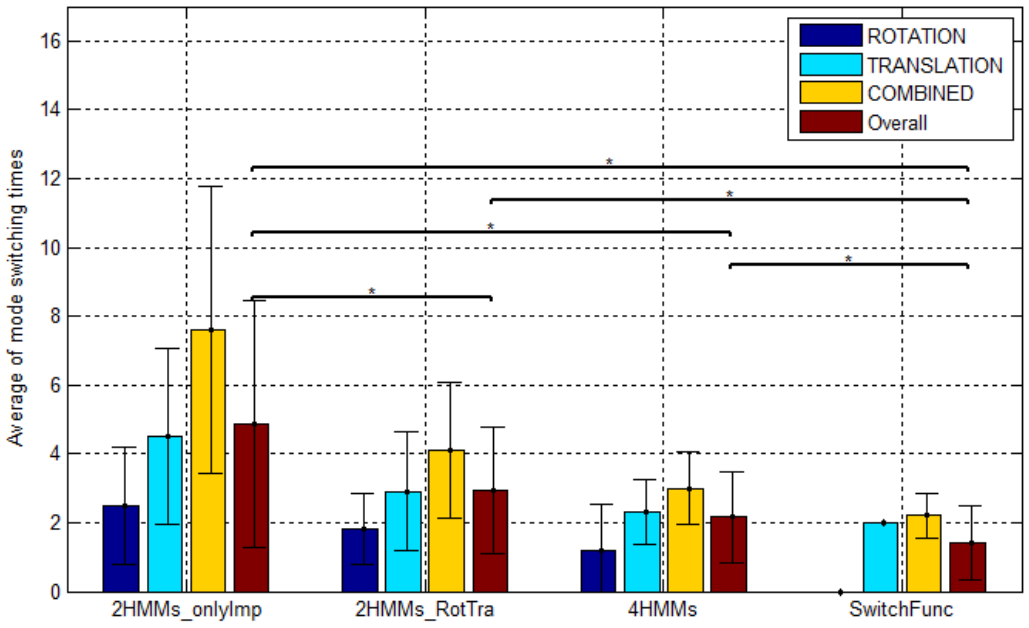


Figure 33: Number of switching mode with different methods. Error bars represent $\pm$ deviation of number of switches across trials. Stars indicate significance of $p < 0.05$

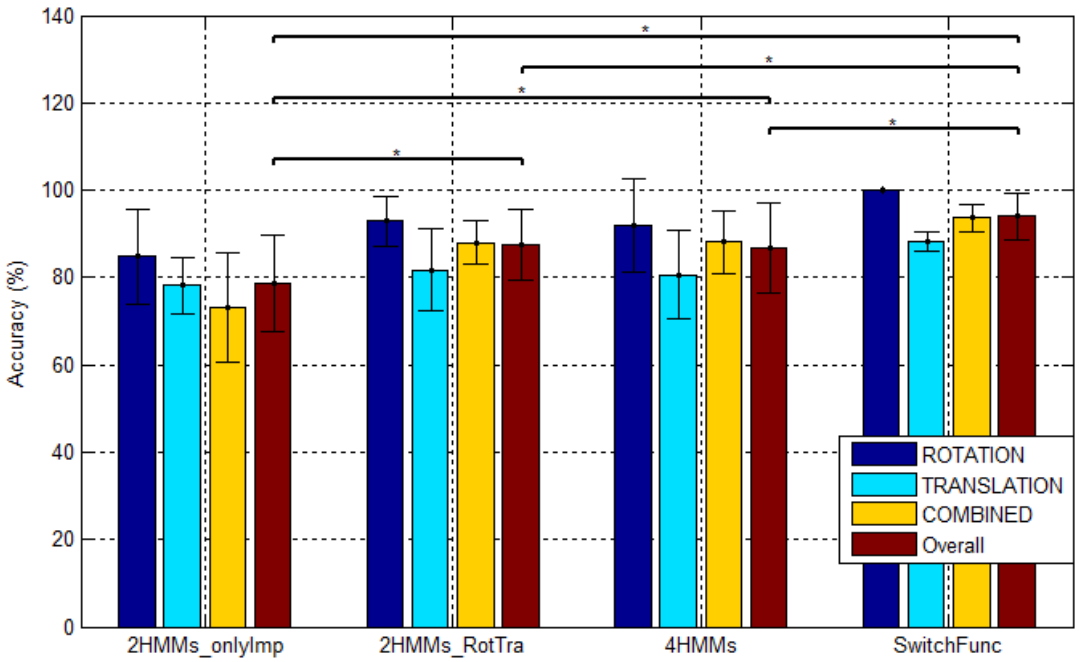


Figure 34: Comparision of accuracies of recognition using different methods. Error bars represent $\pm$ deviation of accuracy across trials. Stars indicate significance of $p < 0.05$

A notice point is that the number of mode switching tend to be inversely proportional to the accuracies (Figure 33, Figure 34). This is understandable because when the accuracy is low, the system will make more wrong predictions and the number of switches will increase. In task COMBINED, as the path is longer and involves both translation and rotation, the number of switches is higher in comparison to task ROTATION and TRANSLATE. The accuracy of HMMs using data collected during impedance control only is significantly lowest ($p < 0.05$) (Figure 34). This is because the training data was collected when the robot ran without defining rotation and translation. Later the trained HMMs are used for recognizing the person's intent and switching mode. The behavior of the robot now depends on the intent mode, which is different from when running in only impedance control without switching mode. Therefore, the performance of recognition decreases. The methods using data collected in each mode of rotation and translation solved the above problem, so that the accuracies of 2HMMs_RotTra and 4HMMs methods are higher and there are less switches. The average of accuracy of 2HMMs_RotTra method is slightly higher

than 4HMMs method. However, there is no statistical significance between these methods. The switching function based on force magnitude has highest accuracy and least number of switches ($p < 0.05$) (Figure 33, Figure 34).

With the swiching function, the person figured out how translation and rotation switches work and performed well in the tests as there was no restriction on speed of movement for each test. However, when specific requirements of fast rotation or slow translation were required, the switching function based on force magnitude performed poorly while HMMs methods still worked well (Figure 35). This is because rather than using only force magnitude information, HMMs used the combination of different signals with both magnitudes and signs which may also contains different informations like magnitude and direction. In addition, the person's movement is better described in time series data which HMMs methods work well with rather than only instantaneous data. The average accuracy of HMMs methods in fast rotation and slow translation is lower than the tests at self-selected speeds (Figure 34), the reason may be HMMs are trained with data at self-selected speed and may be optimized to work at this speed. Then, when these special cases occur, accuracy decreases.

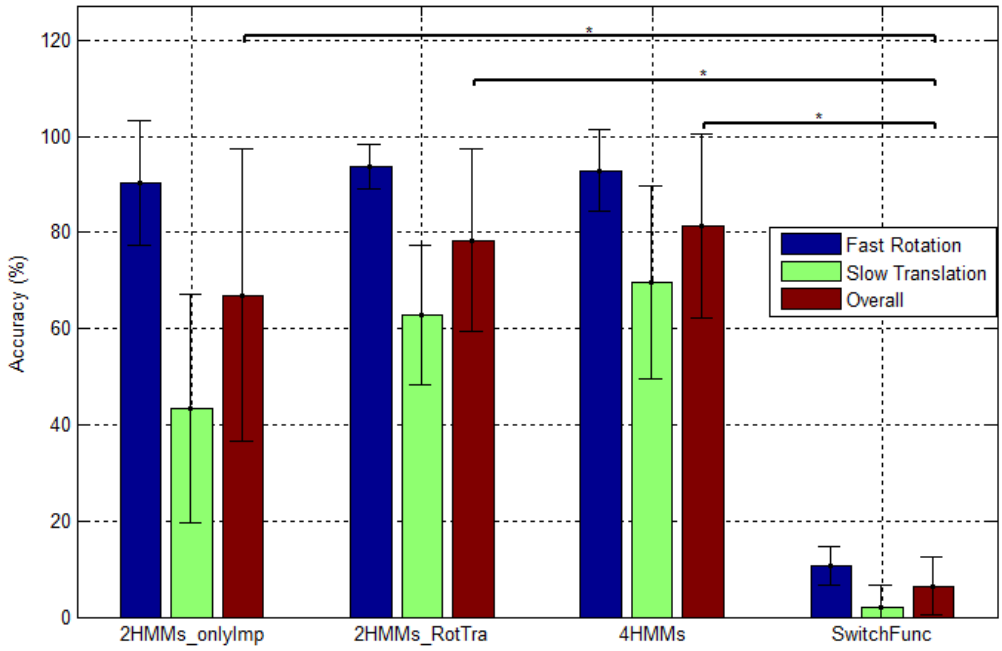


Figure 35: Accuracy in fast rotation and slow translation. Error bars represent $\pm$ deviation of accuracy across trials. Stars indicate significance of $p < 0.05$

CHAPTER 6

CONCLUSION AND FUTURE WORKS

This project has shown that using HMMs can be used for intent recognition of rotation versus translation problem. Using only data from the interaction forces and torques at the robot's end effector, the HMMs were 87.5% accurate in estimating the person's intent. Using the underlying impedance controller, the robot worked smoothly with a person holding the object with one hand in the collaborative manipulation tasks. In this project, the collaborative manipulation tasks are accomplished without using extra force and torque sensors as in in most of prior works (Dumora et al., 2013)(Wojtara et al., 2009). For comparison, the project also implemented the switching based function on force magnitude as proposed in (Karayiannidis et al., 2014). Although the accuracy was higher when using force magnitude as switching criterion at self-selected speeds ($p < 0.05$), HMMs outperformed the switching function in fast rotations and slow translations ($p < 0.05$).

First, this project used a single user and future works with the experimental tests of more people could be conducted to give the better assessment of the method. Second, some future works could improve the accuracies of HMMs for intent recognition. In this project, the force and torque data used as inputs of HMMs are only raw data, so one may look into further methods of extracting features from these data before feeding into HMMs. In addition, the external force data is calculated through the robot positions and the torque sensors at every joint. If only one of these sensors has an error, the data may contains error, which may decrease the performance of HMMs. Therefore, an improvement would be a use of a multiple degree of freedom force and torque sensors attached to the robot's end-effector, so that external force and torque can be measured directly. Third, in this project, HMMs is used for classification of rotation about the robot gripping point and translation laterally in horizontal plane. Future works could implement HMMs for more intents in 3D space manipulation to make the collaboration

completed such as lifting up or down, rotating about the person gripping points, rotating around circular with different diameters. The robot in future works rather than only stands at a fixed position, can be able to move around so that the paths of manipulation are not limited. Finally, an improvement for this project would be implementing the robot with assistant mode. In this project's concept, the robot plays the main role as a follower in cooperation and the user always play the leader role and needs to apply a certain force to move the object. A future concept in which the robot is able to play the role of leader during a move could be applied. Once the robot detects the human partner's movement intent, it can move in advance so that the person does not need to keep applying the force to the object. The person then only needs to exert force when they want to switch the moving paths from rotation to translation or vice versa.

APPENDIX

LINKS FOR UPLOADED FILES

The KUKA program of this project is uploaded at https://github.com/vinhqnguyen/Kuka-Lbr-iiwa-collaboration-using-HMMs- , and the HMM library witen in Java - Jahmm library can be downloaded at https://code.google.com/archive/p/jahmm/

REFERENCES

Al-jarrah, O., & Zheng, Y. (1997). Arm-Manipulator Coordination for Load Sharing using Reflexive Motion Control. *International Conference on Robotics and Automation*, (April), 2326–2331. http://doi.org/10.1109/ROBOT.1997.619309

Arai, H., Takubo, T., Hayashibara, Y., & Tanie, K. (2000). Human-robot cooperative manipulation using a virtual nonholonomic\nconstraint. *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, *4*(April), 4063–4069.

Buerger, S. P., & Hogan, N. (2007). Actuation system with fluid transmission for interaction control and high force haptics. Retrieved from https://www.google.com/patents/US7284374

Chen, F.-S., Fu, C.-M., & Huang, C.-L. (2003). Hand gesture recognition using a real-time tracking method and hidden Markov models. *Image and Vision Computing*, *21*(8), 745–758. http://doi.org/10.1016/S0262-8856(03)00070-2

Connectivity, K. S. (2014). KUKA Sunrise . Connectivity Servoing 1 . 5.

Dumora, J., Geffard, F., Bidard, C., Aspragathos, N. a., & Fraisse, P. (2013). Robot assistance selection for large object manipulation with a human. *Proceedings - 2013 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2013*, 1828–1833. http://doi.org/10.1109/SMC.2013.315

Feth, D., Groten, R., Peer, A., Hirche, S., & Buss, M. (2009). Performance Related Energy Exchange in Haptic Human-Human Interaction in a Shared Virtual Object Manipulation Task. *Joint EuroHaptics Conf. and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 338––343.

Flanagan, J., & Johansson, R. (2002). Hand Movements. *Encyclopedia of the Human Brain*, *2*, 399–414.

Gribovskaya, E., Kheddar, A., & Billard, A. (2011). Motion learning and adaptive impedance for robot control during physical interaction with humans. *Proceedings - IEEE International Conference on Robotics and Automation*, 4326–4332. http://doi.org/10.1109/ICRA.2011.5980070

Hogan, N. (1985). Impedance Control : An Approach to Manipulation :, (March).

Hogan, N. (2011). Interaction control.

Hogan, N., & Buerger, S. P. (2005). Impedance and Interaction Control 19.1. *Robotics and Automation Handbook*, 19–1. Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Impedance+and+Interaction+Control#0

Huang, X. D. (1992). Phoneme classification using semicontinuous hidden Markov models. *IEEE Transactions on Signal Processing*, *40*(5), 1062–1067. http://doi.org/10.1109/78.134469

Ikeura, R., & Inooka, H. (1995). Variable impedance control of a robot for cooperation with a human. *Proceedings - IEEE International Conference on Robotics and Automation*, *3*, 3097–3102. http://doi.org/10.1109/ROBOT.1995.525725

Juang, R. B. H. (1986). An Introduction to Hidden Markov Models, (January).

Karayiannidis, Y., Smith, C., & Kragic, D. (2014). Mapping human intentions to robot motions via physical interaction through a jointly-held object. *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, 391–397. http://doi.org/10.1109/ROMAN.2014.6926284

Kucukyilmaz, A., Sezgin, T. M., & Basdogan, C. (2013). Intention recognition for dynamic role exchange in haptic collaboration. *IEEE Transactions on Haptics*, *6*(1), 58–68. http://doi.org/10.1109/TOH.2012.21

Lawitzky, M., Mortl, A., & Hirche, S. (2010). Load Sharing in Human-Robot Cooperative Manipulation. *Proc. {IEEE Ro-Man}*, 1–6.

Os, K. S., & Workbench, K. S. (2014). KUKA Sunrise.OS 1.5 KUKA Sunrise.Workbench 1.5.

Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*. http://doi.org/10.1109/5.18626

Rozo, L., Calinon, S., Caldwell, D., Jimenez, P., Torras, C., & Jiménez, P. (2013). Learning Collaborative Impedance-based Robot Behaviors. *AAAI Conference on Artificial Intelligence*, 1422–1428.

Samur, E. (2012). *Springer Series on Touch and Haptic Systems: Performance Metrics for Haptic Interfaces*.

Starner, T. E. (1995). Visual Recognition of American Sign Language Using Hidden Markov Models., *298*. Retrieved from http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA344219

Stefanov, N., Passenberg, C., Peer, A., & Buss, M. (2013). Design and evaluation of a haptic computer-assistant for telemanipulation tasks. *IEEE Transactions on Human-Machine Systems*, *43*(4), 385–397. http://doi.org/10.1109/TSMC.2013.2257743

Stefanov, N., Peer, A., & Buss, M. (2010). Online Intention Recognition for Computer-Assisted Teleoperation. *Proc. {IEEE ICRA}*, 5334–5339.

Takeda, T., Kosuge, K., & Hirata, Y. (2005). HMM-based dance step estimation for dance partner robot - MS DanceR-. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 1602–1607. http://doi.org/10.1109/IROS.2005.1545207

Wang, Z., Peer, A., & Buss, M. (2009). An HMM approach to realistic haptic Human-Robot interaction. *Proceedings - 3rd Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, World Haptics 2009*, 374–379. http://doi.org/10.1109/WHC.2009.4810835

Wojtara, T., Uchihara, M., Murayama, H., Shimoda, S., Sakai, S., Fujimoto, H., & Kimura, H. (2009). Human–robot collaboration in precise positioning of a three-dimensional object. *Automatica*, *45*(2), 333–342. http://doi.org/10.1016/j.automatica.2008.08.021

Yi, L., & Zhang, J. (2001). Fundamental Limitations and Differences of Robust and Adaptive Control I, 4802–4807.

Yigit, S., Burghart, C., & Worn, H. (2004). Co-operative carrying using pump-like constraints. *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, *4*, 3877–3882 vol.4. http://doi.org/10.1109/IROS.2004.1390019

Yokoyama, K., Handa, H., Isozumi, T., Fukase, Y., Kaneko, K., Kanehiro, F., … Hirukawa, H. (2003). Cooperative works by a human and a humanoid robot. *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, *3*, 2985–2991. http://doi.org/10.1109/ROBOT.2003.1242049

Jean-Marc Francois, Willem V. Onsem: Jahmm library (https://code.google.com/archive/p/jahmm/downloads)